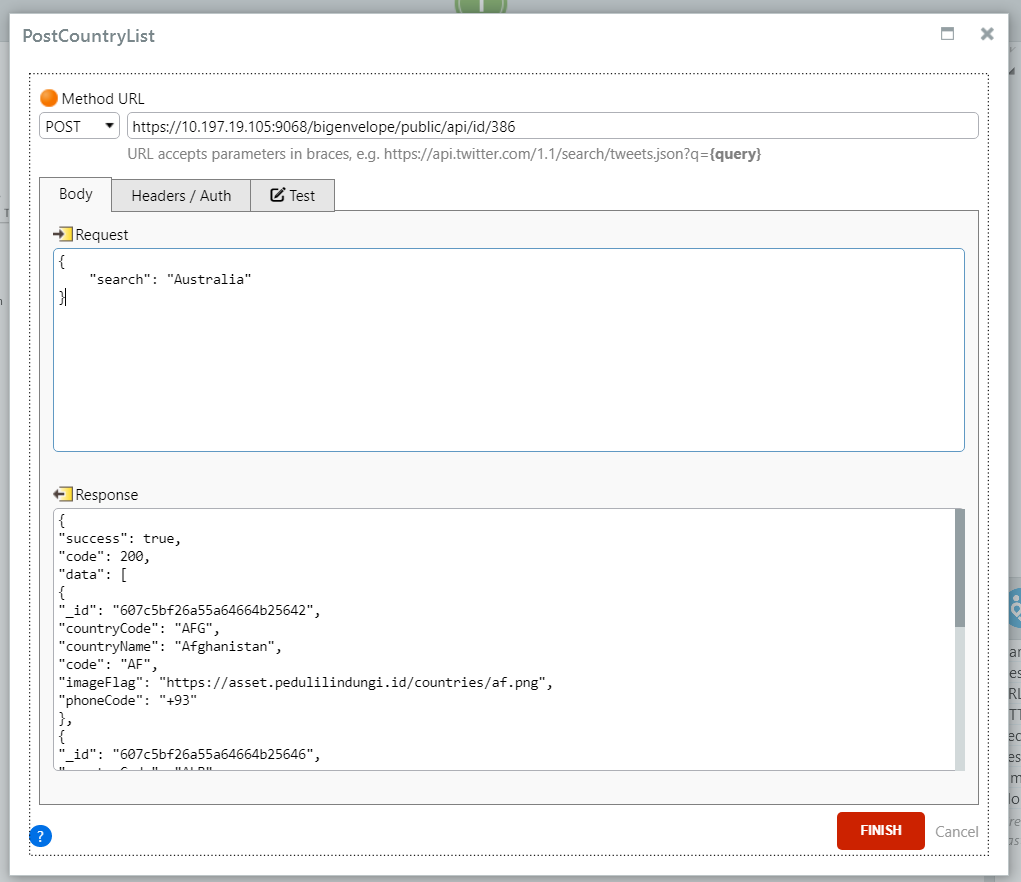
1. Integrasi PeduliLindungi
2. Services
3. PeduliLindungi REST API

Terdapat 5 REST API yang disediakan oleh tim PeduliLindungi sesuai dengan dokumen integrasi.

* PostCountryList

Berfungsi untuk mendapatkan list negara untuk kebutuhan API PostVerificationId nomor passport.



Request Header:

|  |  |  |
| --- | --- | --- |
| Header | Type | Example |
| Authorization | Text |  |
| BDI-Key | Text |  |
| ReferenceNumber | Number | 1234567 |

Request Body:

{

"search": "A"

}

Response Body:

{

"success": true,

"code": 200,

"data": [

{

"\_id": "607c5bf26a55a64664b25642",

"countryCode": "AFG",

"countryName": "Afghanistan",

"code": "AF",

"imageFlag": "https://asset.pedulilindungi.id/countries/af.png",

"phoneCode": "+93"

},

{

"\_id": "607c5bf26a55a64664b25646",

"countryCode": "ALB",

"countryName": "Albania",

"code": "AL",

"imageFlag": "https://asset.pedulilindungi.id/countries/al.png",

"phoneCode": "+355"

},

{

"\_id": "607c5bf46a55a64664b25681",

"countryCode": "DZA",

"countryName": "Algeria",

"code": "DZ",

"imageFlag": "https://asset.pedulilindungi.id/countries/dz.png",

"phoneCode": "+213"

}

],

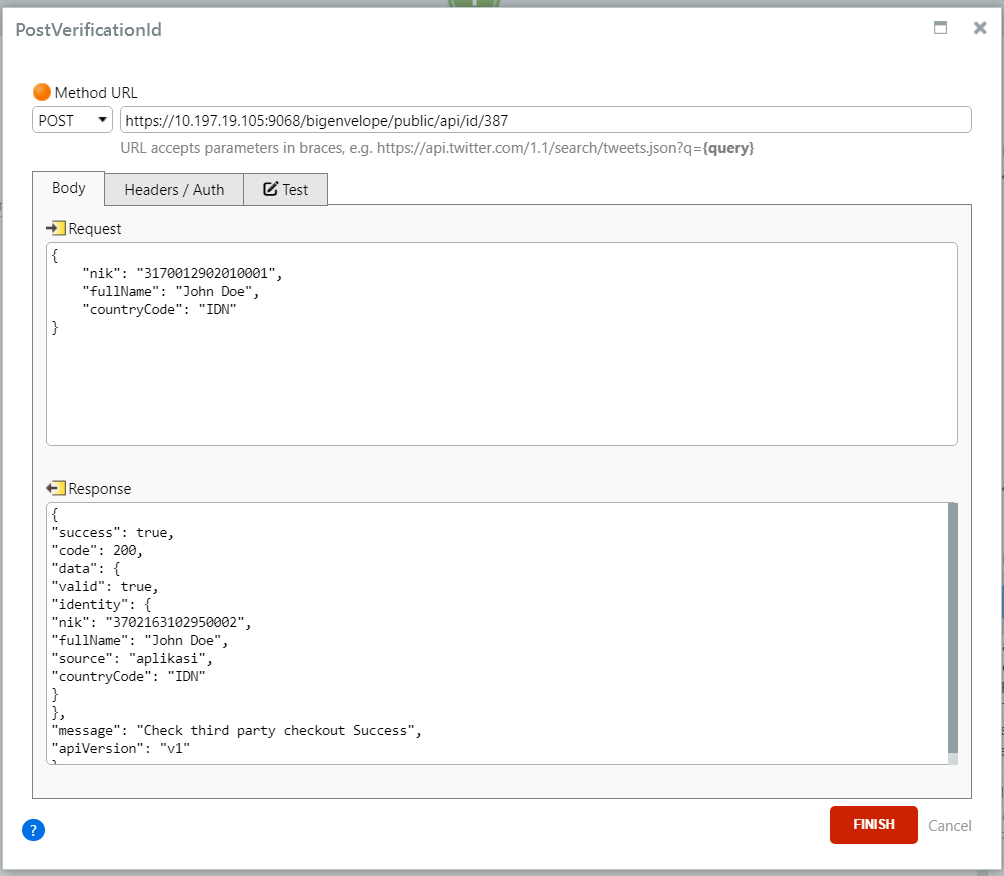
"message": "Get countries success",

"apiVersion": "v1"

}

* PostVerificationId

Berfungsi untuk verifikasi KTP berdasarkan NIK dan Nama.



Request Header:

|  |  |  |
| --- | --- | --- |
| Header | Type | Example |
| Authorization | Text |  |
| BDI-Key | Text |  |
| ReferenceNumber | Number | 1234567 |

Request Body:

{

"nik": "3170012902010001",

"fullName": "John Doe",

"countryCode": "IDN"

}

Response Body:

{

"success": true,

"code": 200,

"data": {

"valid": true,

"identity": {

"nik": "3702163102950002",

"fullName": "John Doe",

"source": "aplikasi",

"countryCode": "IDN"

}

},

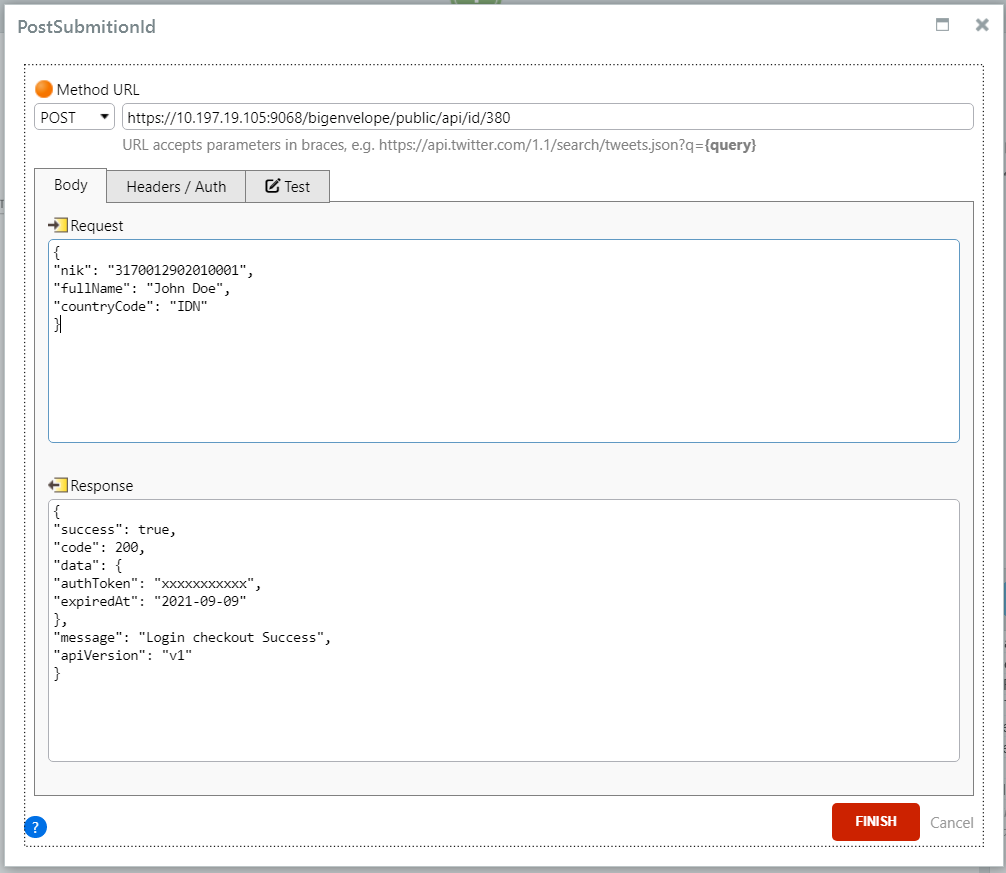
"message": "Check third party checkout Success",

"apiVersion": "v1"

}

* PostSubmitionId

Berfungsi untuk submit data NIK dan Nama untuk mendapatkan BearerCode untuk kebutuhan preview scan.



Request Header:

|  |  |  |
| --- | --- | --- |
| Header | Type | Example |
| Authorization | Text |  |
| BDI-Key | Text |  |
| ReferenceNumber | Number | 1234567 |

Request Body:

{

"nik": "3170012902010001",

"fullName": "John Doe",

"countryCode": "IDN"

}

Response Body:

{

"success": true,

"code": 200,

"data": {

"authToken": "xxxxxxxxxxx",

"expiredAt": "2021-09-09"

},

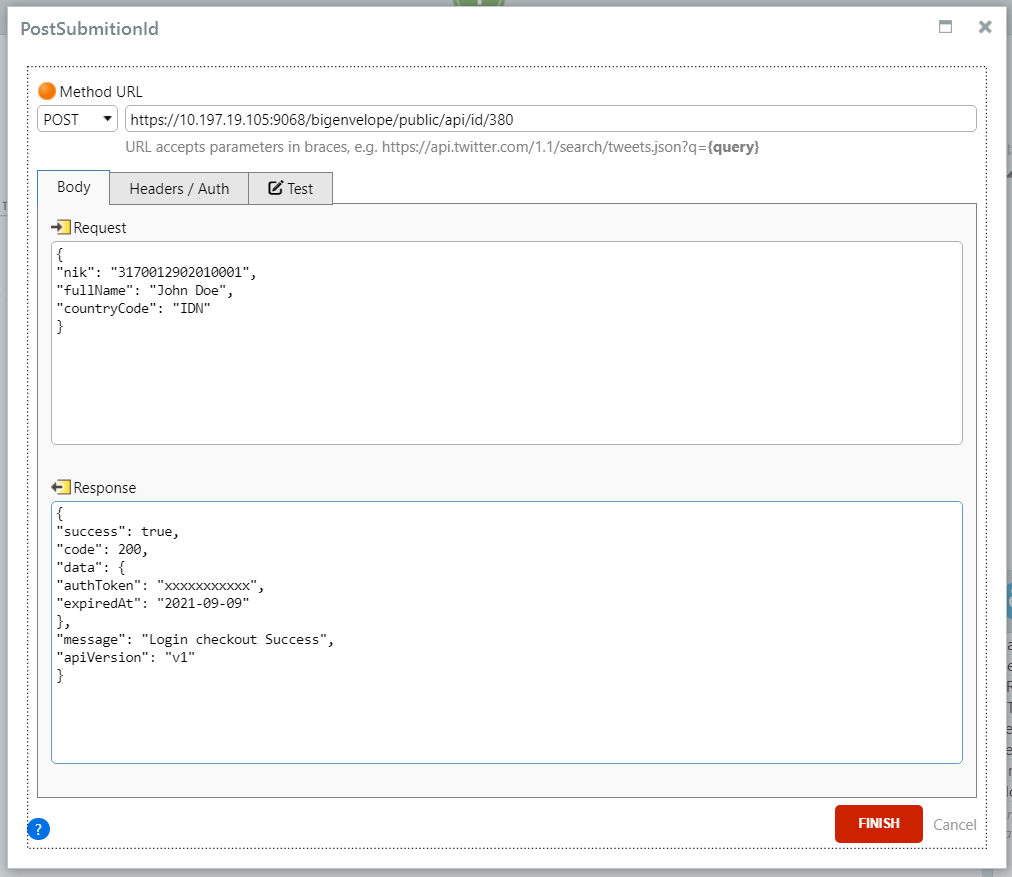
"message": "Login checkout Success",

"apiVersion": "v1"

}

* PostPreviewScan

Berfungsi menampilkan data tempat yang terdaftar di PeduliLindungi menggunakan isi QRCode dari PeduliLindungi.



Request Body:

{

"qrCode": "checkin:60dd9e2bac68e40011056662"

}

Response Body:

{

"success": true,

"code": 200,

"data": {

"name": "Camp Nou",

"category": {

"key": "mall",

"name": "Mall",

"image": {

"fileUrl": "https://dev-asset.pedulilindungi.id/assets/place-gedung.png"

}

},

"activityCategory": "indoor",

"location": {

"address": "Jawa Timur, 66217",

"longitude": 0,

"latitude": 0,

"provinceId": "35",

"cityId": "3504",

"districtId": "350401"

},

"checkoutPin": {

"latitude": -8.068494,

"longitude": 111.8992385,

"radius": 200

},

"image": {

"fileUrl": "https://dev-asset.pedulilindungi.id/place.jpeg"

},

"crowd": 1,

"maxCapacity": 100,

"checkpoint": {

"name": "Pintu Depan",

"checkinpin": {

"latitude": -8.068494,

"longitude": 111.8992385,

"radius": 50

},

"checkoutpin": {

"latitude": 0,

"longitude": 0,

"radius": 0

},

"qrCode": {

"checkIn": "checkin:60dd9e2bac68e40011056662",

"checkOut": "checkout:60dd9e2bac68e40011056662"

}

}

},

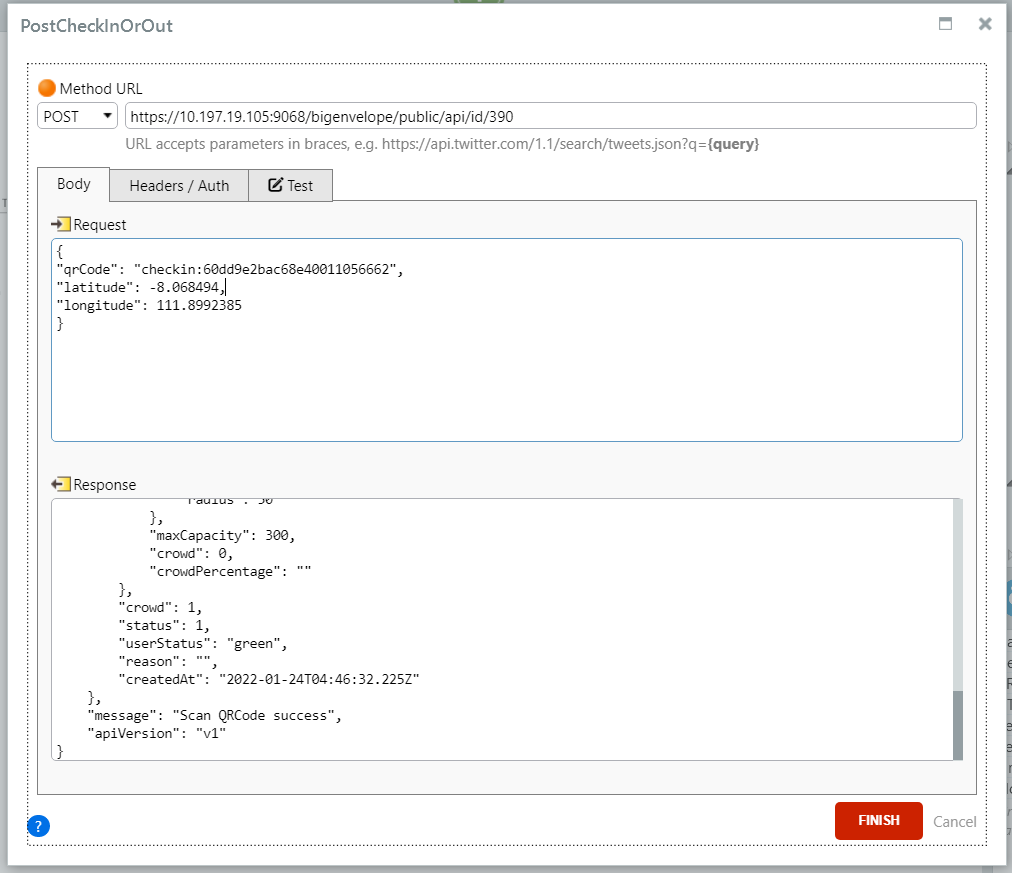
"message": "Get place ticket detail",

"apiVersion": "v1"

}

* PostCheckInOrOut

Berfungsi untuk memvalidasi QrCode Check In atau Out.



Request Body:

{

"qrCode": "checkin:60dd9e2bac68e40011056662",

"latitude": -8.068494,

"longitude": 111.8992385

}

Response Body:

{

"success": true,

"code": 200,

"data": {

"id": "61ee2f282fd112115e896816",

"checkInTime": "2022-01-24T11:46:32.225+07:00",

"checkOutTime": "2022-01-24T12:05:07.860479868+07:00",

"place": {

"name": "Mall Nusa Indah",

"activityType": "indoor",

"checkPointName": "Pintu Depan edit",

"category": {

"key": "mall",

"name": "Mall",

"active": true,

"image": {

"fileUrl": "https://dev-asset.pedulilindungi.id/assets/place-gedung.png"

}

},

"location": {

"address": "Tebet",

"longitude": 111.8992385,

"latitude": -8.068494,

"districtId": "350216",

"districtName": "",

"cityId": "3502",

"cityName": "",

"provinceId": "35",

"provinceName": ""

},

"image": {

"fileUrl": "https://asset.pedulilindungi.id/place/place-image-611333e75eb61f008ebf7483.jpeg"

},

"qrCode": {

"checkIn": "checkin:611333e75eb61f008ebf7485",

"checkOut": "checkout:611333e75eb61f008ebf7485"

},

"checkoutPin": {

"latitude": -8.068494,

"longitude": 111.8992385,

"radius": 50

},

"maxCapacity": 300,

"crowd": 0,

"crowdPercentage": ""

},

"crowd": 1,

"status": 1,

"userStatus": "green",

"reason": "",

"createdAt": "2022-01-24T04:46:32.225Z"

},

"message": "Scan QRCode success",

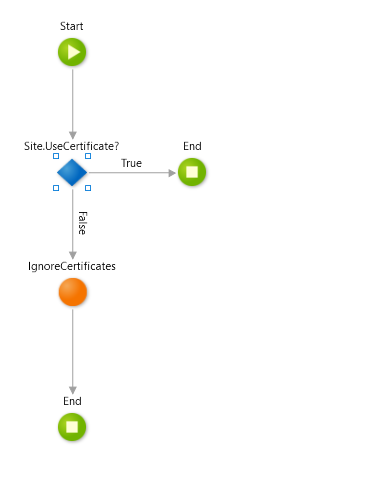
"apiVersion": "v1"

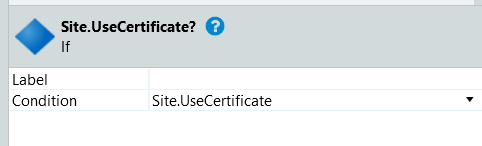
}

1. Consume Services in Danamon Care

* OnBeforeRequestAdvanced

Action untuk pengecekan apakah REST API menggunakan valid certificate atau tidak

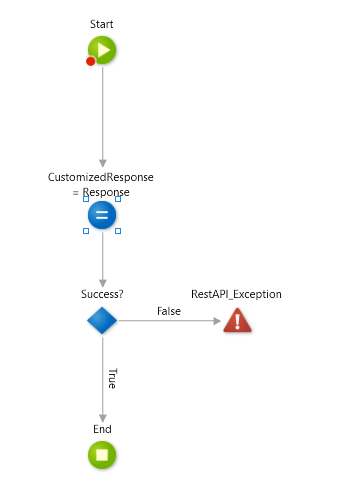


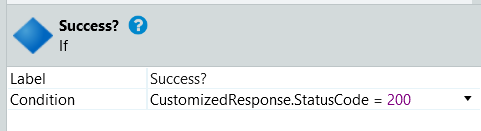


Pengecekan value site properties UseCertificate = True/False

* OnAfterResponse

Action pemeriksaan value status Response API



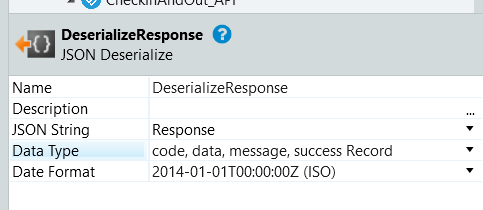


Pengecekan StatusCode = 200

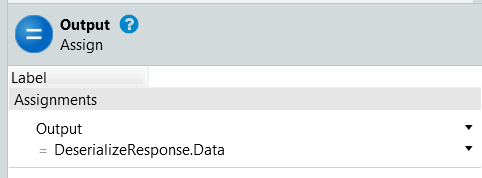
* DeseriliazeResponse

Deserialize json text dari Exception dari OnAfterResponse.

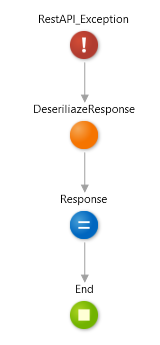
|  |  |  |
| --- | --- | --- |
| Response | Input Parameter | Text |
| Output | Output Parameter | Object |
| Code | Output Parameter | Integer |
| Data | Output Parameter | Text |
| Message | Output Parameter | Text |
| Success | Output Parameter | Boolean |



Deserialize Response dari Input Parameter ke Format JSON



Assign hasil deserialize ke object Output

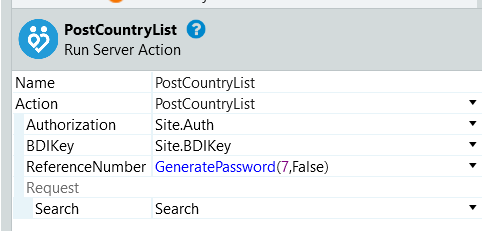


Pada Exception tiap Server Action membutuhkan deserialize response dari REST API, agar message yang ditampilan bukan dalam tipe Text.

* CountryList

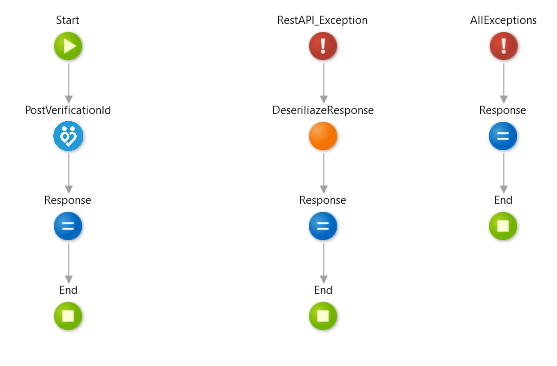
Action untuk mengambil data dari REST API PostCountryList

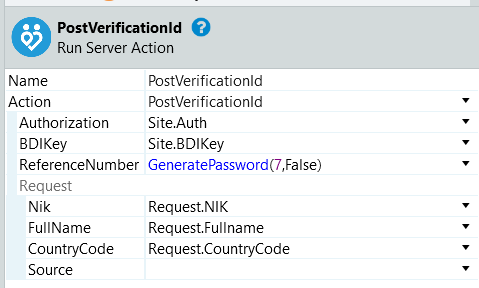




* VerificationId

Action untuk mengambil data dari REST API PostVertificationId

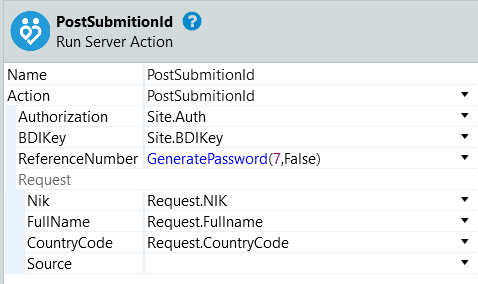




* SubmitionId

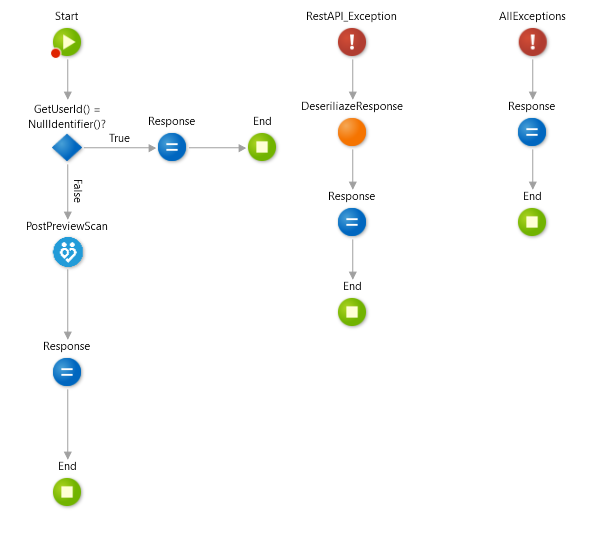
Action untuk mengambil data dari REST API PostSubmitionId

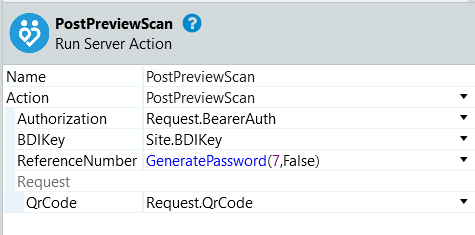




* PreviewScan

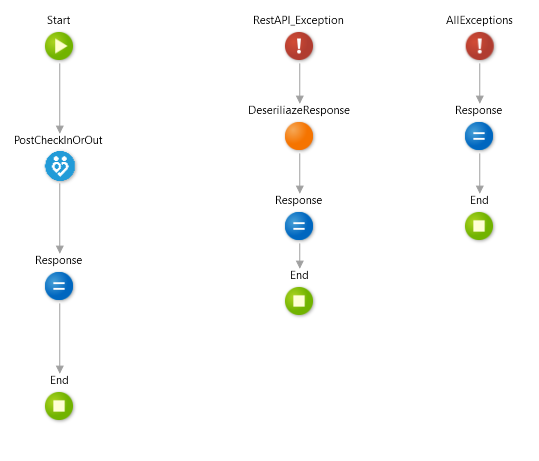
Action untuk mengambil data dari REST API PostPreviewScan

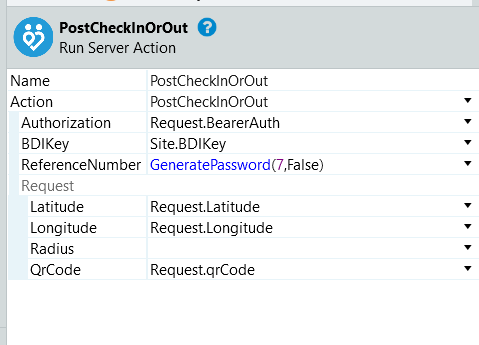




* CheckInOrOut

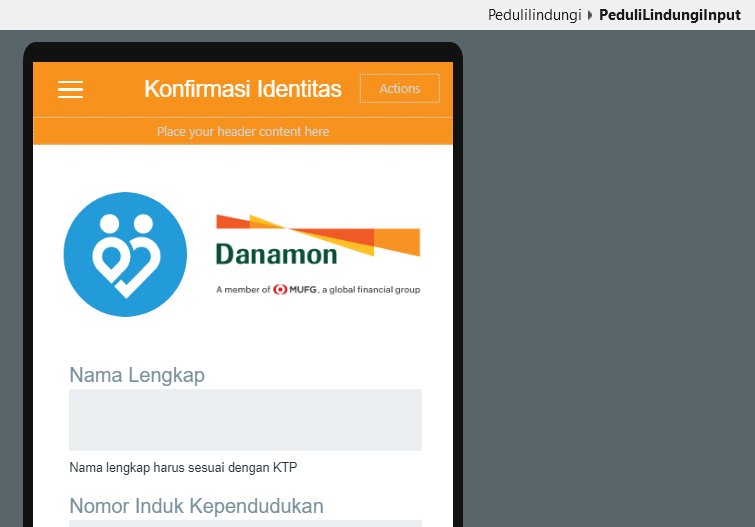
Action untuk mengambil data dari REST API PostCheckInOrOut





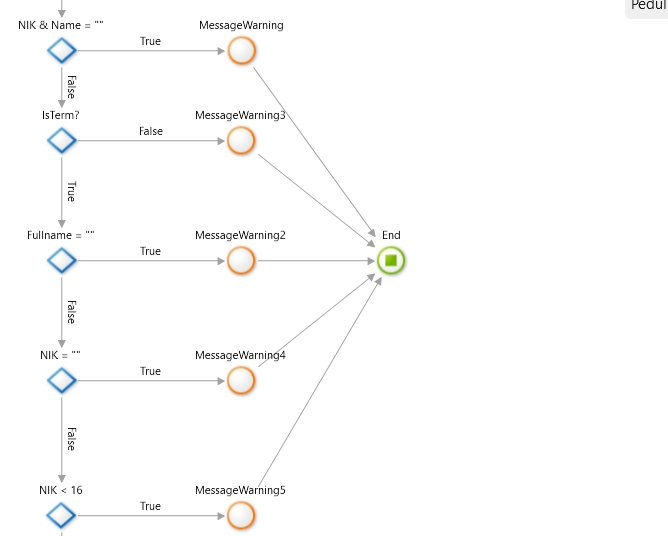
1. Consume PL Services on Danamon Care

* PeduliLindungiInput

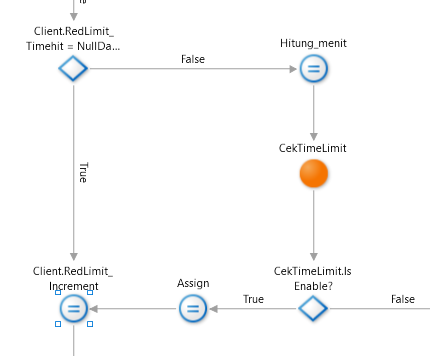


Berikut adalah tampilan pada halaman PeduliLindungiInput. Terdapat 2 Input (Nama Lengkap dan Nomor Induk Kependudukan) dan 1 Button.

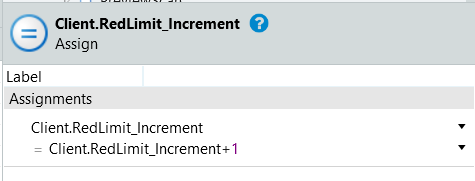
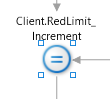
Pada fungsi button Lanjut Safe Entrance, terdapat flow client validasi.



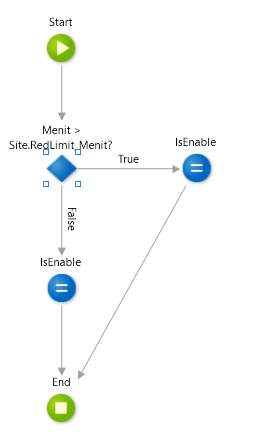
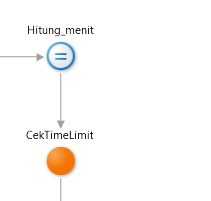
Lalu setelah validasi, akan ada pengecekan limit penggunaan PeduliLindungiInput. Pada if Client.RedLimit\_Timehit = NullDate() adalah pengecekan untuk apakah user sudah pernah melakukan input pada halaman ini.



Apabila belum (true branch) akan ada penambahan limit increment.

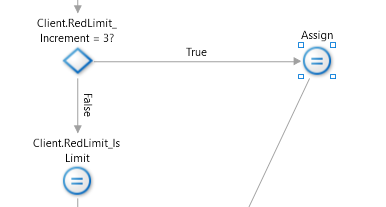


Apabila sudah (false branch) akan ada assignment pengecekan sudah berapa menit user melakukan input.

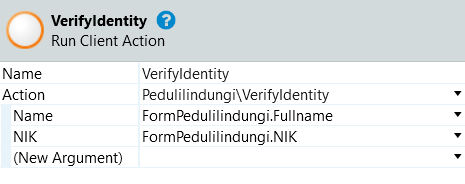
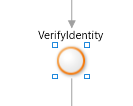


Lanjut ke true branch Client.RedLimit\_Timehit = NullDate() sesudah assignment limit increment.

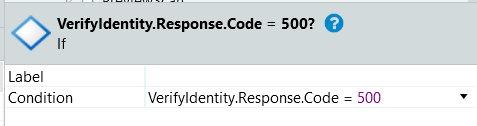
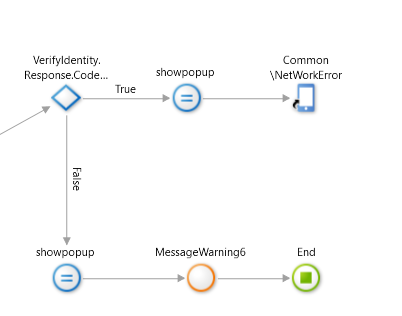
Terdapat pengecekan apakah increment sudah sama dengan 3.



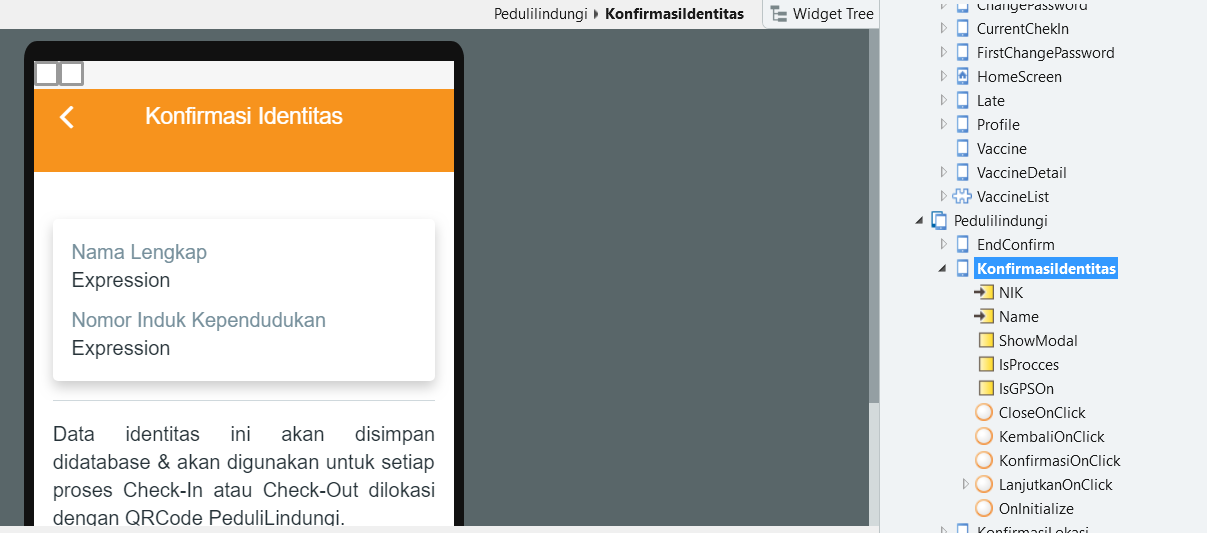
Apabila berhasil melalui validasi, terdapat client action yang mengconsume service PeduliLindungi untuk API VerificationId.



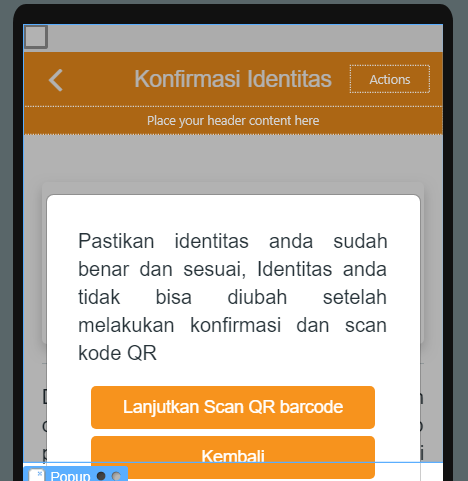
Apabila Success = True akan navigate ke screen berikutnya. Apabila Success = False akan muncul warning yang menunjukan error message dari API PeduliLindungi. Terdapat 1 tambahan kondisi untuk pengecekan Response Code = 500. Apabila response code yang dibalikan dari PeduliLindungi = 500 maka akan diarahkan ke screen network error.



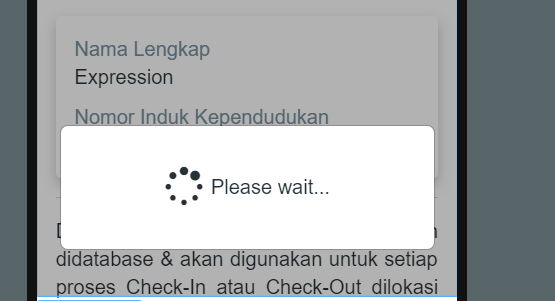
* KonfirmasiIdentitas



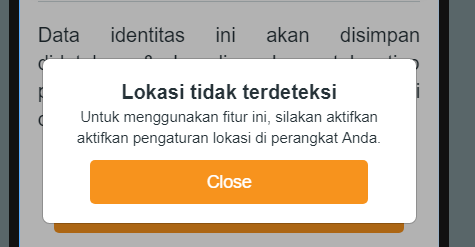
Berikut adalah tampilan pada halaman KonfirmasiIdentitas. Terdapat 2 input parameter yaitu NIK dan Name. Lalu ada 3 variable ShowModal, IsProcess dan IsGPSOn.



ShowModal berfungsi untuk menunjukan pop up confirmasi untuk lanjut scan QR Barcode atau kembali.

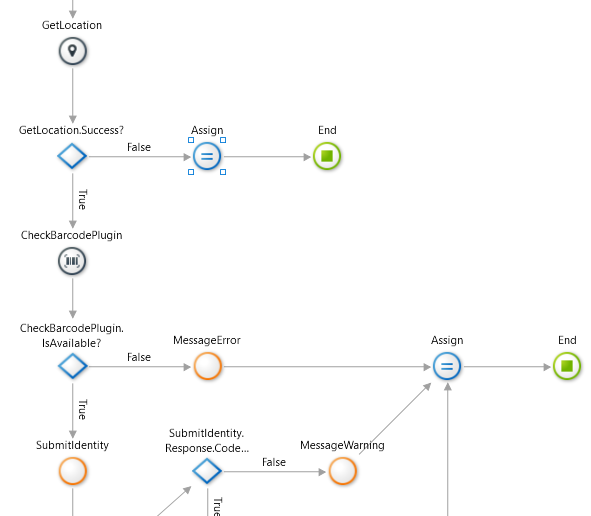


IsProcess untuk menunjukan loading.

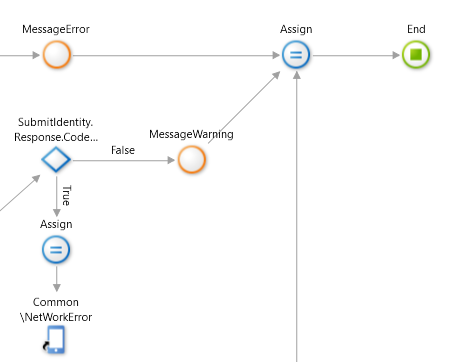


isGPSOn untuk menunjukan notifikasi lokasi tidak terdeteksi.

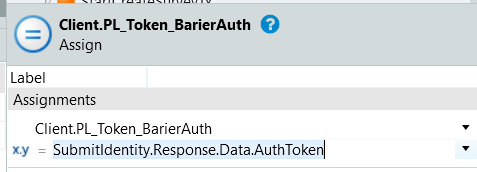
Terdapat Client Action LanjutkanOnClick yang terhubung ke Button pada popup Konfirmasi (ShowModal). Pada LanjutkanOnClick terdapat validasi pengecekan LocationPlugin. Apabila GetLocation.Success = False maka akan memunculkan popup lokasi tidak terdeteksi. Sebaliknya, jika True maka akan dilanjutkan untuk hit API SubmitionId pada client action SubmitIdentity.



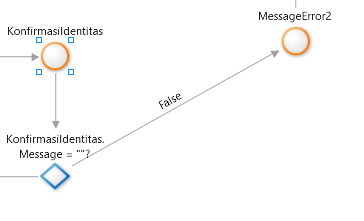
Apabila Success = False akan muncul warning yang menunjukan error message dari API PeduliLindungi. Terdapat 1 tambahan kondisi untuk pengecekan Response Code = 500. Apabila response code yang dibalikan dari PeduliLindungi = 500 maka akan diarahkan ke screen network error.

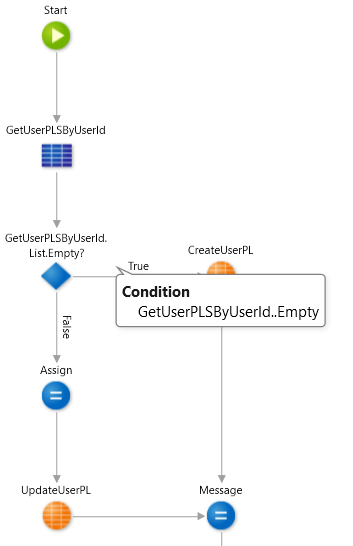


Apabila Success = True akan ada assignment Client.PL\_Token\_BarierAuth = SubmitIdentity.Response.Data.AuthToken.



Setelah itu terdapat client action KonfirmasiIdentitas yang isinya ada pengecekan apakah user tersebut sudah pernah melakukan konfirmasi identitas PeduliLindungi.

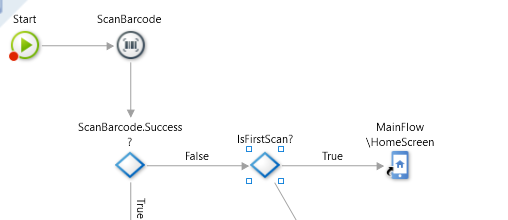




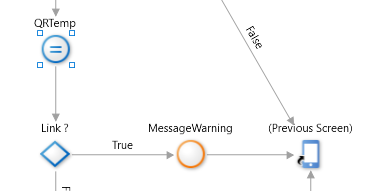
Apabila success akan pindah ke halaman berikutnya.

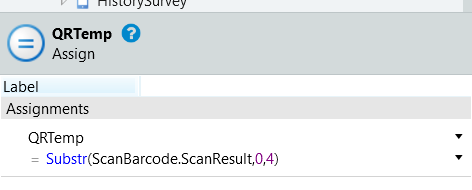
* PreviewScan

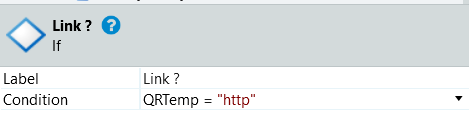
Tampilan PreviewScan hanya kosongan. Halaman ini difungsikan untuk menampilkan scanner barcode. Untuk menampilkan scanner, terdapat client action OnReady. Pada action ini terdapat validasi pengecekan success scan barcode.



Terdapat validasi pengecekan barcode bukan barcode url.

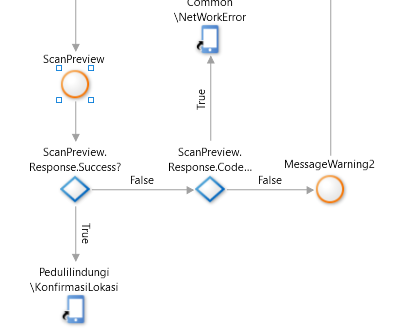






Setelah validasi, terdapat client action ScanPreview untuk menjalankan API ScanPreview.

Apabila Success = False akan muncul warning yang menunjukan error message dari API PeduliLindungi. Terdapat 1 tambahan kondisi untuk pengecekan Response Code = 500. Apabila response code yang dibalikan dari PeduliLindungi = 500 maka akan diarahkan ke screen network error.

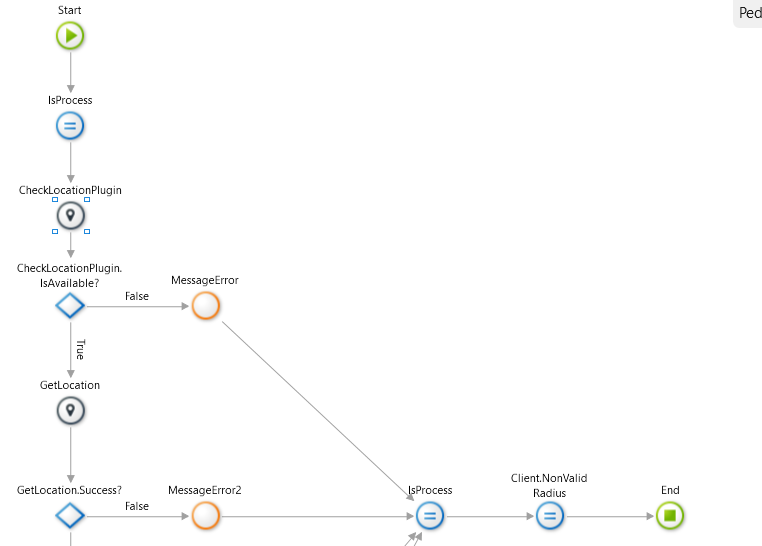


* KonfirmasiLokasi

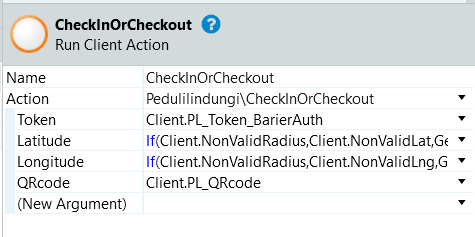
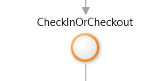
Pada halaman ini, terdapat beberapa input parameter. ImageURL, Gate, PlaceName, AddressPlace, ActivityCategory, Crowd, MaxCapacity. Parameter-parameter ini merupakan response value yang didapatkan pada API ScanPreview untuk dapat ditampilan dihalaman KonfirmasiLokasi.



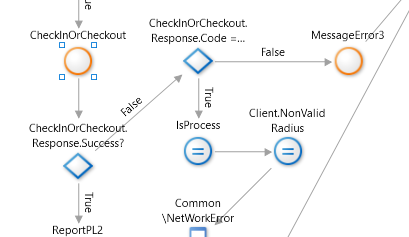
Terdapat validasi location plugin pada button CheckIn/CheckOut, pada client action ConfirmOnClick.



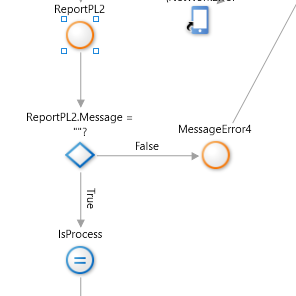
Setelah validasi tersebut, dilakukan pengeckan CheckIn atau CheckOut pada client action CheckInOrOut.



Apabila Success = False akan muncul warning yang menunjukan error message dari API PeduliLindungi. Terdapat 1 tambahan kondisi untuk pengecekan Response Code = 500. Apabila response code yang dibalikan dari PeduliLindungi = 500 maka akan diarahkan ke screen network error.



Setelah itu, dilakukan penginputan log report PL pada client action ReportPL.



* EndConfirm

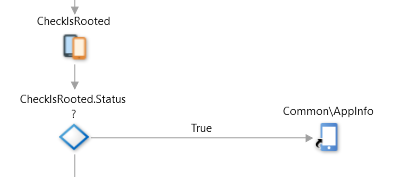
Pada halaman ini hanya menampilkan data response dari API sebelumnya yaitu API CheckInOrOut. Terdapat beberapa parameter yang dibutuhkan untuk menampilkan data-data pada halaman ini.



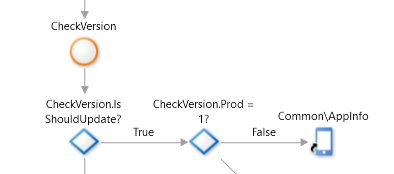
1. Update Features

Pada halaman Common > Splash terdapat sebuah fitur pengecekan versi yang digunakan untuk direct pengguna ke halaman Google Play Store. Fitur ini terdapat pada Client Action OnLoadComplete.

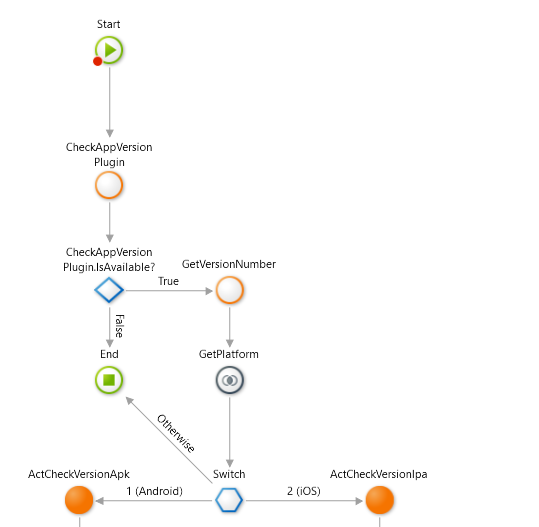
Sebelum ke pengecekan versi, terdapat validasi pengecekan apakah handphone pengguna adalah hp root atau bukan.



Apabila status bukan Rooted. Maka akan dilanjutkan ke client action CheckVersion.

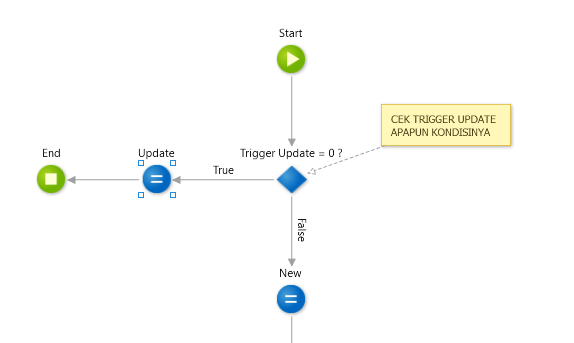


Di dalam action ini akan dilakukan pengecekan plugin AppVersionPlugin. Apabila berhasil GetVersionNumber, flow akan diarahkan ke switch case statement berdasarkan hasil action GetPlatform.

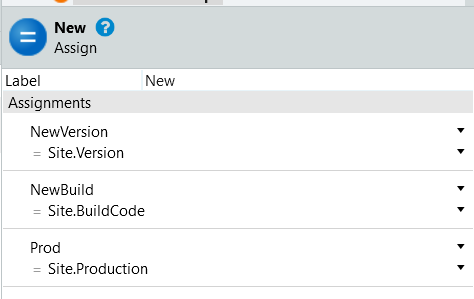


Statement pertama, apakah platform handphone pengguna adalah android, maka akan diarahkan ke ActCheckVersionAPK. Sebaliknya, jika handphone pengguna adalah ios akan diarahkan ke ActCheckVersionIpa. Kedua fungsi ini sama, hanya berbeda dipenggunaan Site Propertiesnya.

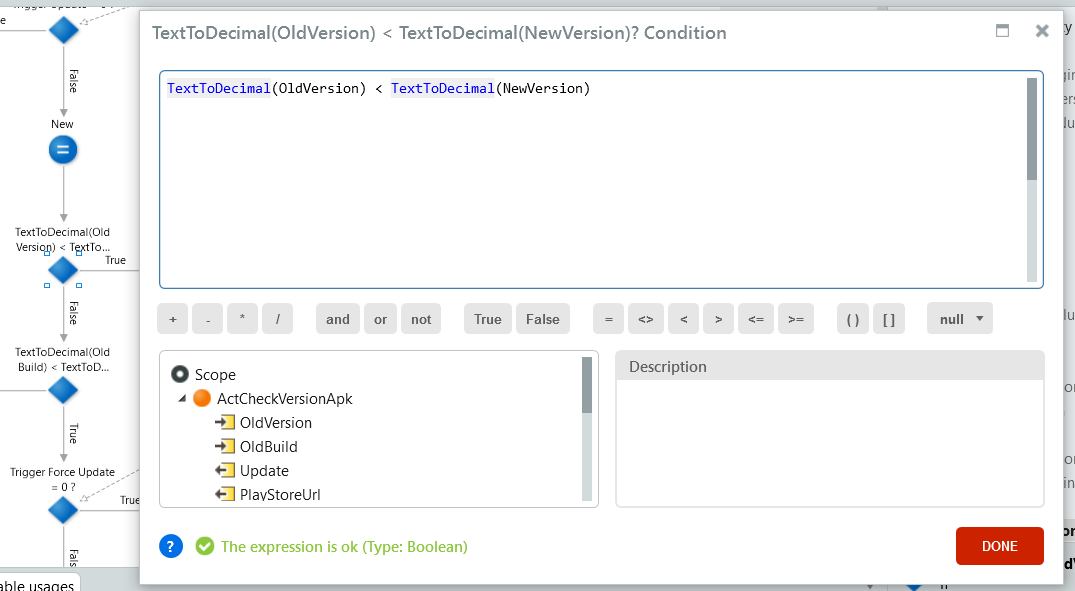
Pada action ActCheckVersionApk/Ipa terdapat validasi apakah site properties update sudah dinyalakan, jika Trigger Update = 0 maka akan diarahkan ke True Branch yang terdapat assignment variable Update = False.



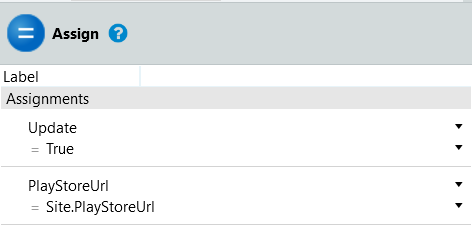
Apabila Trigger Update = 1, akan ada assignment pengambilan value version dari Site Properties (Site.Version, Site. BuildCode dan Site. Production).



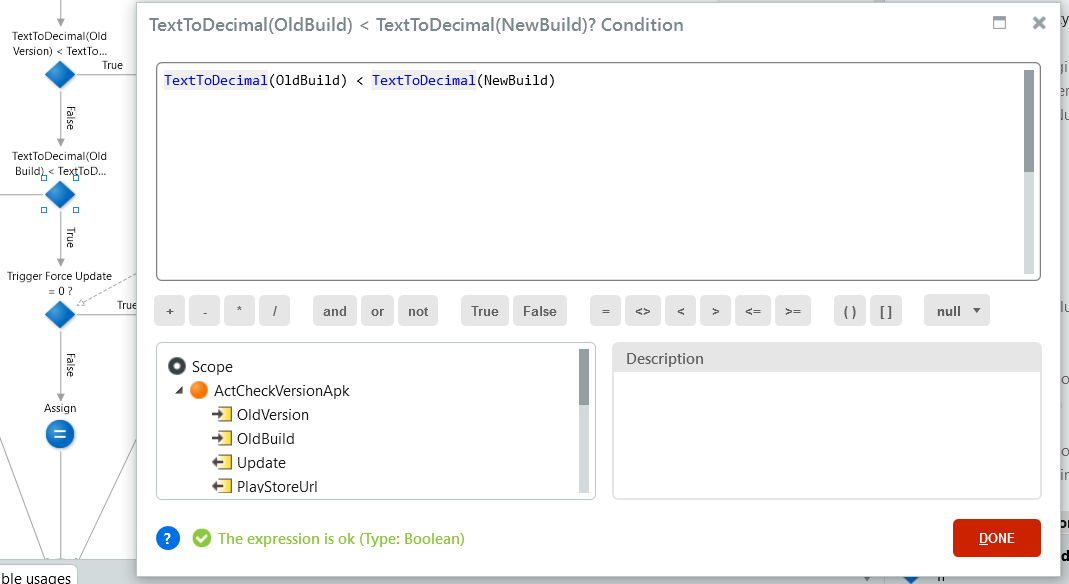
Setelah assignment tersebut, terdapat pengecekan OldVersion < NewVersion.



Apabila hasilnya true, maka ada assignment untuk output parameter Update dan PlayStoreUrl.

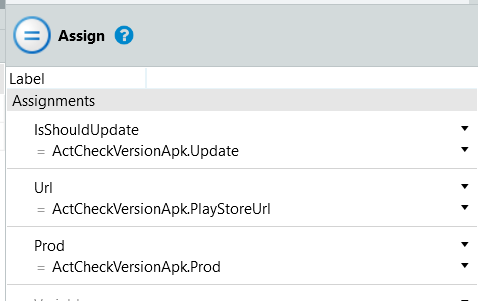


Apabila sebaliknya, akan ada pengecekan OldBuildCode dan NewBuildCode.

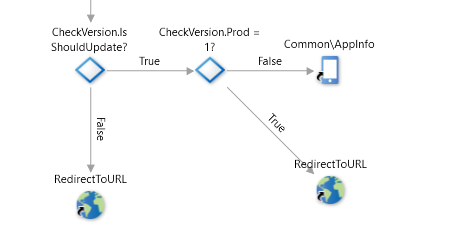


Apabila True, akan diarahkan ke force update, yang terdapat assignment Update dan PlayStoreUrl. Sebaliknya jika false akan diarahkan ke Update = False.

Kembali ke client action CheckVersion, setelah menjalankan Server Action tersebut. Assign ouput parameter ActCheckVersion ke output parameter CheckVersion.



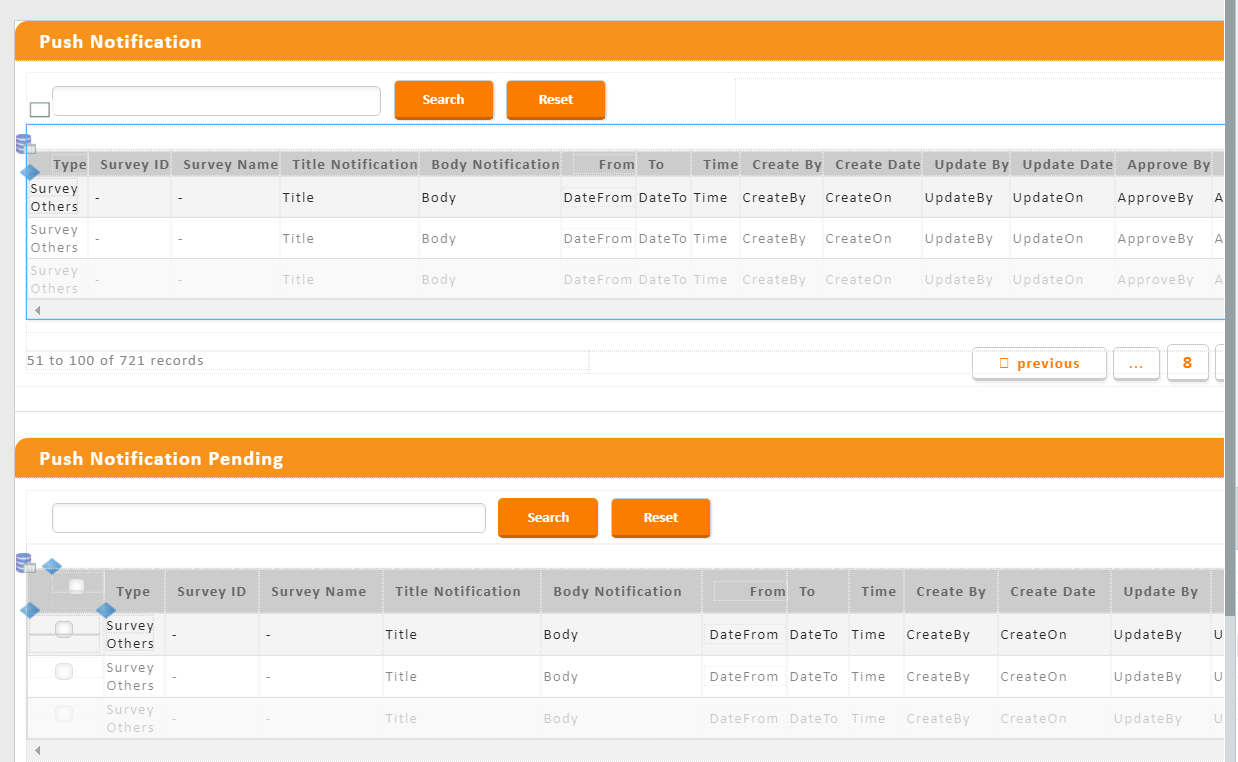
Lalu setelah assign value tersebut, maka pada action di halaman Common > Splash > OnLoadComplete. Terdapat pengecekan IsShouldUpdate = True/False. Apabila false akan diarahkan ke screen login, jika sebalikannya akan diarahkan ke halaman playstore.

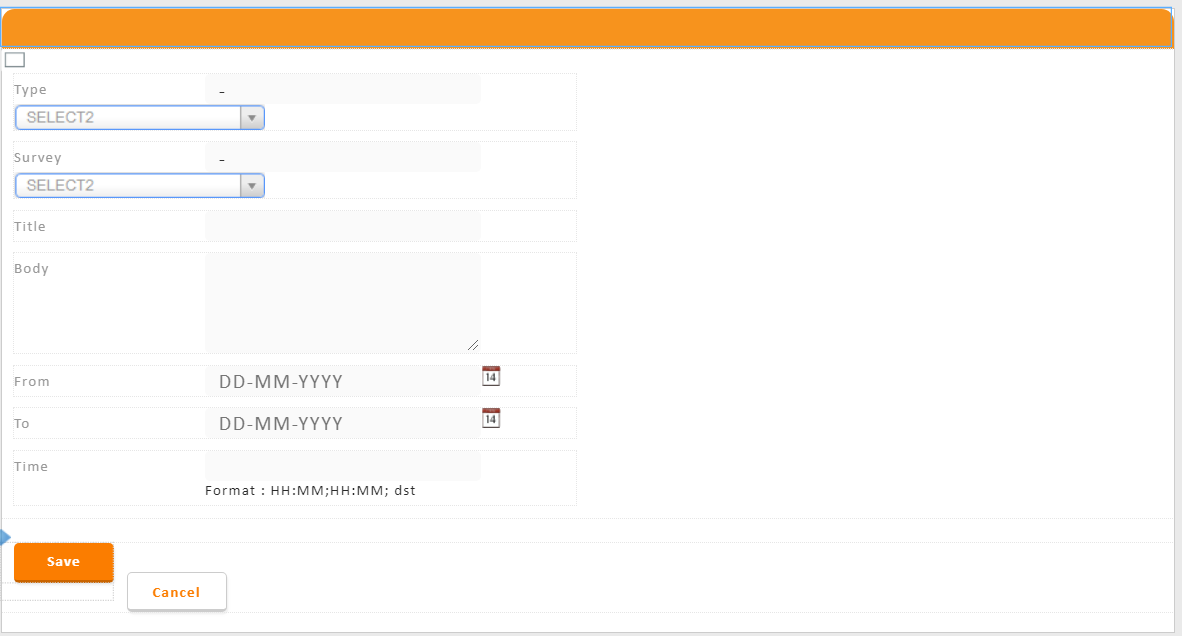


Notes:

1. Site Properties Production merupakan value untuk pengecekan apabila value = 1 maka akan diarahkan langsung ke url PlayStore. Sebaliknya, akan diarahkan ke halaman AppInfo.
2. BuildCode tidak terlalu menjanjikan untuk fitur update, disarankan menggunakan perbandingan version. Untuk site properties yang digunakan sebaiknya site properties Version/VersionIpa, TriggerUpdateAPK/IPA = 1, TriggerForceUpdateAPK/IPA = 0.
3. Scheduler Notifications
4. Notifications Management

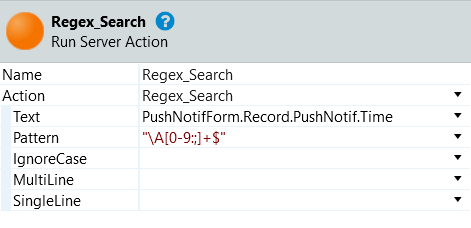
Terdapat perubahan pada back end Notifikasi Survey. Dimana Notifikasi sekarang dibuatkan menu sendiri.



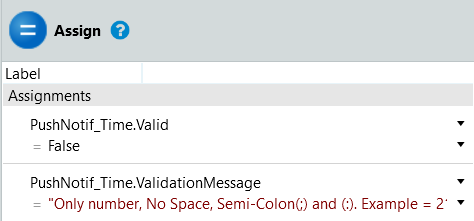


Secara umum, mekanisme checker dan maker adalah sama. Terdapat validasi pada save data push notification yang harus diperhatikan, yaitu validasi format time. Time yang benar adalah dengan format ini HH:MM;HH:MM. Dimana, pemisah antara jam dan menit menggunakan titik dua (:), sedangkan pemisah waktu adalah titik koma (;).

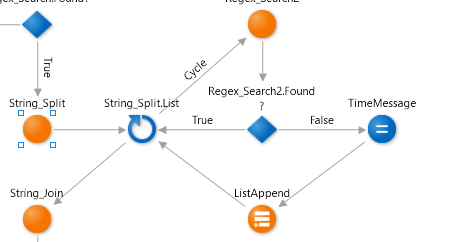
Terdapat validasi pada action Save, dimana dibutuhkan regex\_search untuk mencari patern input field.

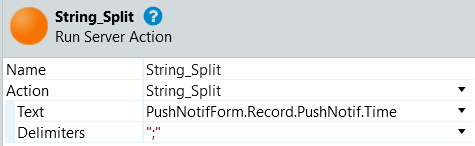


Apabila tidak ditemukan, akan ada assignment ValidationMessage.

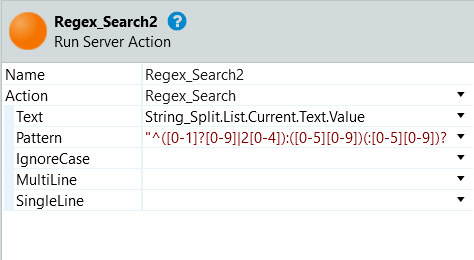


Jika ditemukan, lanjutkan pengecekan value yang diinput sudah sesuai atau belum. Lakukan String\_Split delimeter titik dua untuk mendapatkan list time yang diinput.

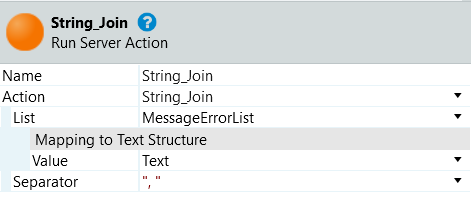
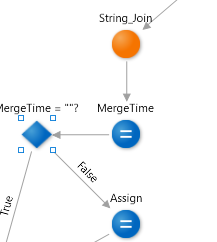


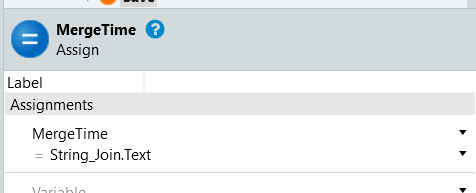


Setelah itu, lalukan regex\_search lagi pada for each, untuk mengecek apakah format timenya sudah sesuai.

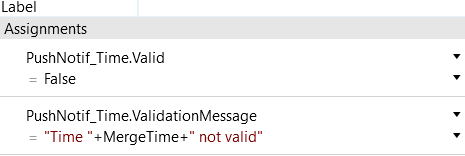


Apabila tidak ditemukan, assign list time ke sebuah varibale list tampungan time error. Sebaliknya, tetap lanjutkan pengecekan for each. Setalah melalukan for each, join list tampungan timer error menggunakan string\_join. Lalu assign value string\_join ke variable MergeTime.



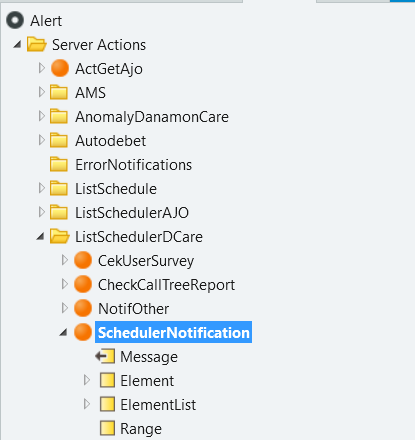


Setelah itu lakukan validasi, apabila MergeTime = “” maka tidak perlu memunculkan ValidationMessage. Sebaliknya, munculkan ValidationMessage.

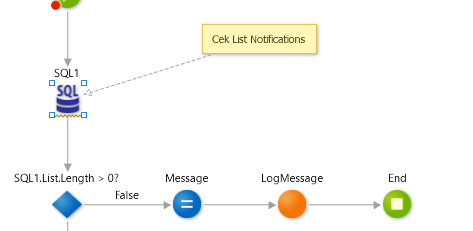


1. Scheduler Notification

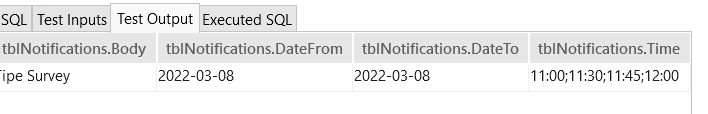
Fitur ini terdapat di Application Alert > Alert > ListSchedulerDcare > Schedule Notification

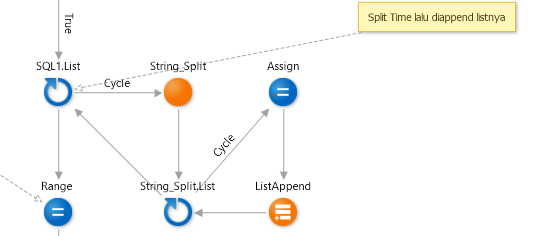


Hal pertama pada flow adalah, diharuskan memeriksa pada table master PushNotif apakah pada tanggal sekarang ada data atau tidak. Sistem akan menjalankan advanced query EXEC [ActivitylogDB].[dbo].[SP\_ScheduleListNotif] @Date. Di mana input @Date merupakan CurrDate().

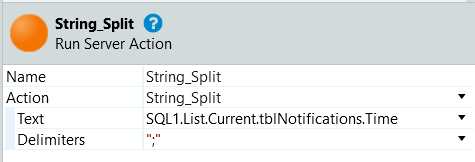


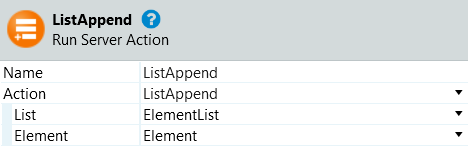
Setelah melakukan pemeriksaan, apabila terdapat data. Hal selanjutnya adalah memisahkan isi kolom Time. Menjadi sebuah list menggunakan for each.





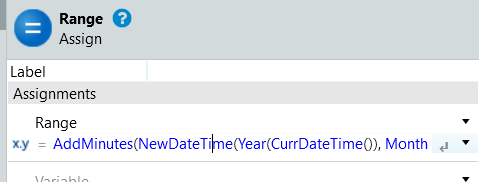
Didalam foreach, lakukan string\_split untuk memisahkan waktu, lalu tampung hasil list string\_split pada variable list ElementList menggunakan ListAppend.



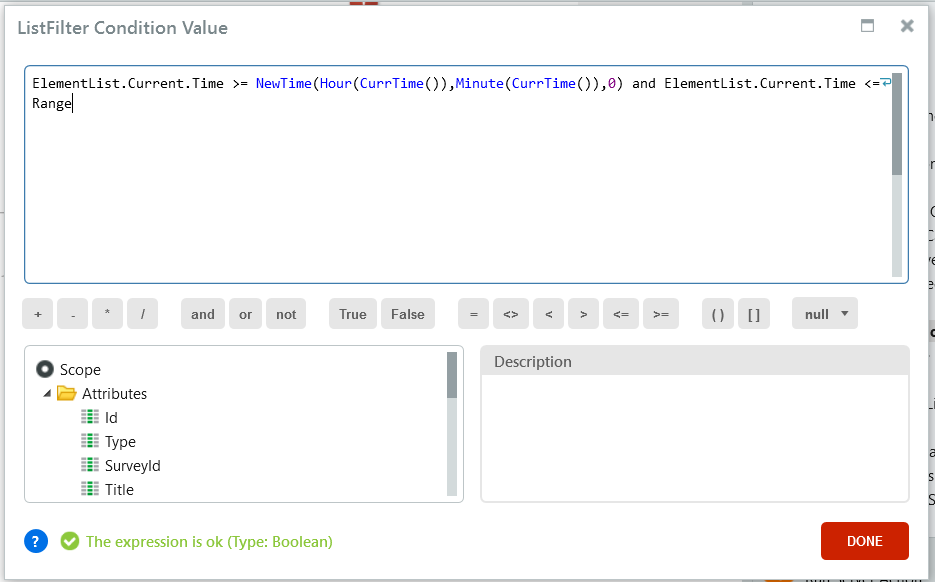


Setelah sukses melakukan for each, definisikan range rentang waktu menggunakan assignment Range = AddMinutes(NewDateTime(Year(CurrDateTime()), Month(CurrDateTime()), Day(CurrDateTime()), Hour(CurrDateTime()), Minute(CurrDateTime()),0)

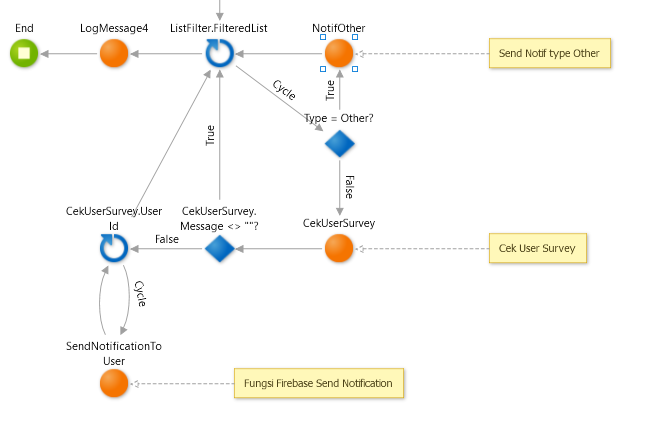
,Site.NotifRangeMinutes). Diharuskan hard code detik/seconds menjadi 0, agar tidak kelebihan detik pada function CurrDateTime().



Setelag itu, filter list ElementList menggunakan ListFilter. ElementList.Current.Time >= NewTime(Hour(CurrTime()),Minute(CurrTime()),0) and ElementList.Current.Time <= Range. Seperti sebelumnya, hal sama yang harus dilakukan adalah, hard code detik/seconds menjadi 0 pada function CurrTime().

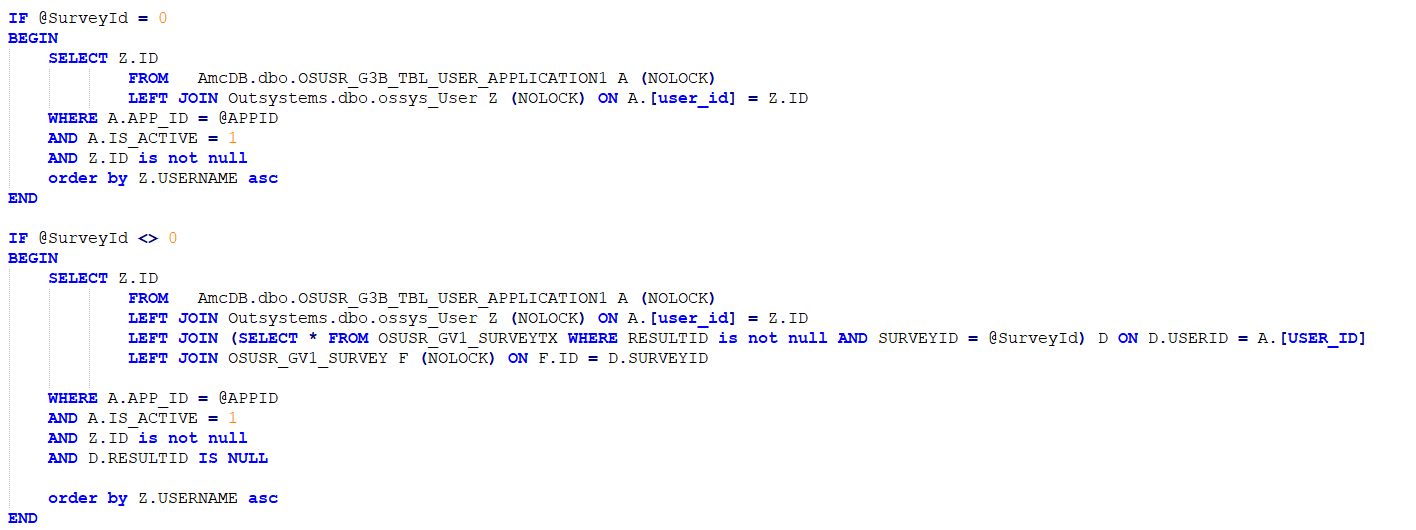


Hal berikutnya adalah mengirim notifikasi menggunakan Firebase. Untuk mengirim notifikasi kita harus melakukan for each pada list yang sudah kita filter tadi. Untuk notif other ada pengecekan pada if, dimana value type pada table master = “Other” atau bukan.



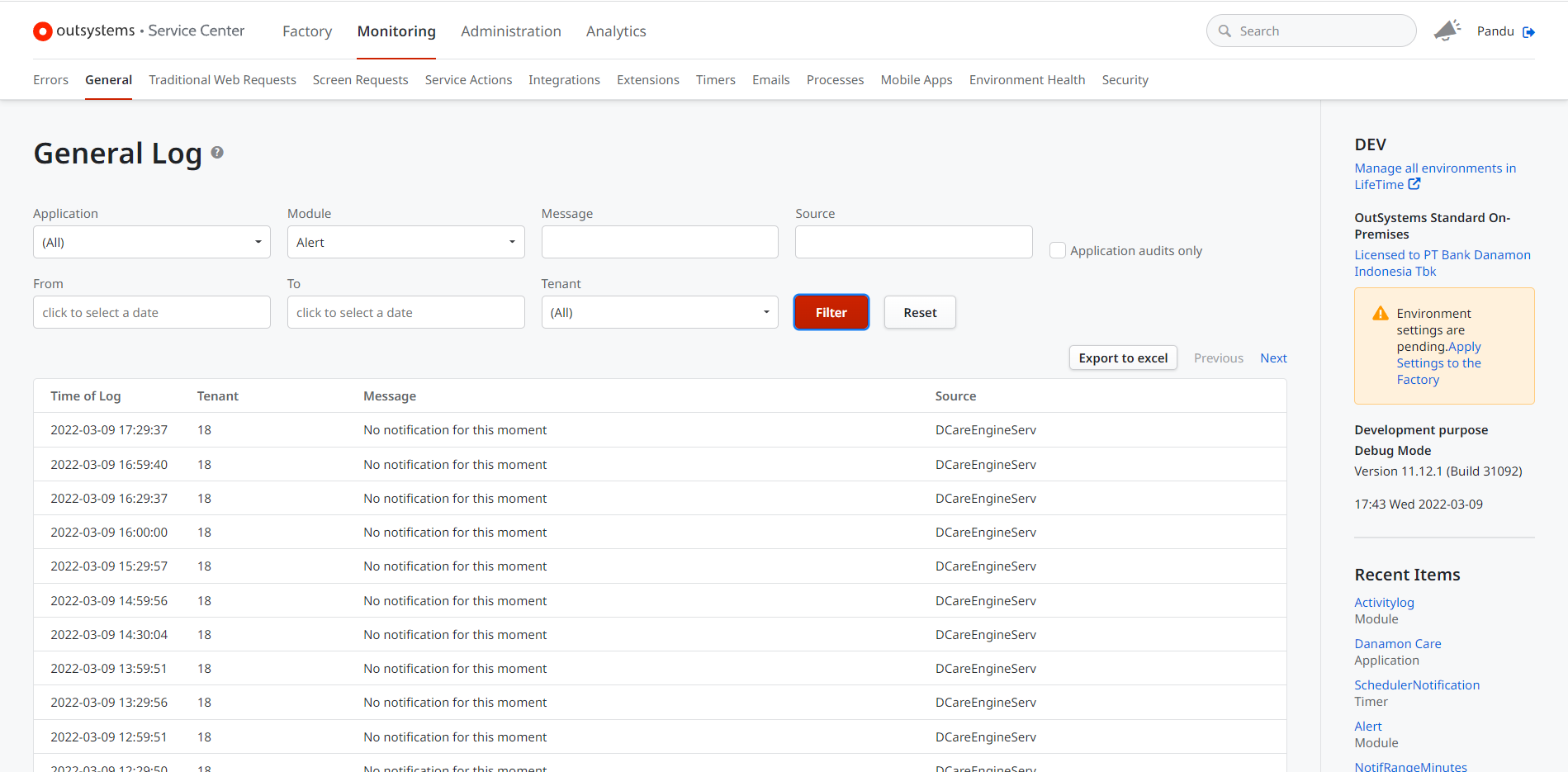
Pada dasarnya, query pada NotifOther dan CekUserSurvey adalah sama. Yang membedakan adalah penggunaan query pada SP SP\_NotifGetAllUser.

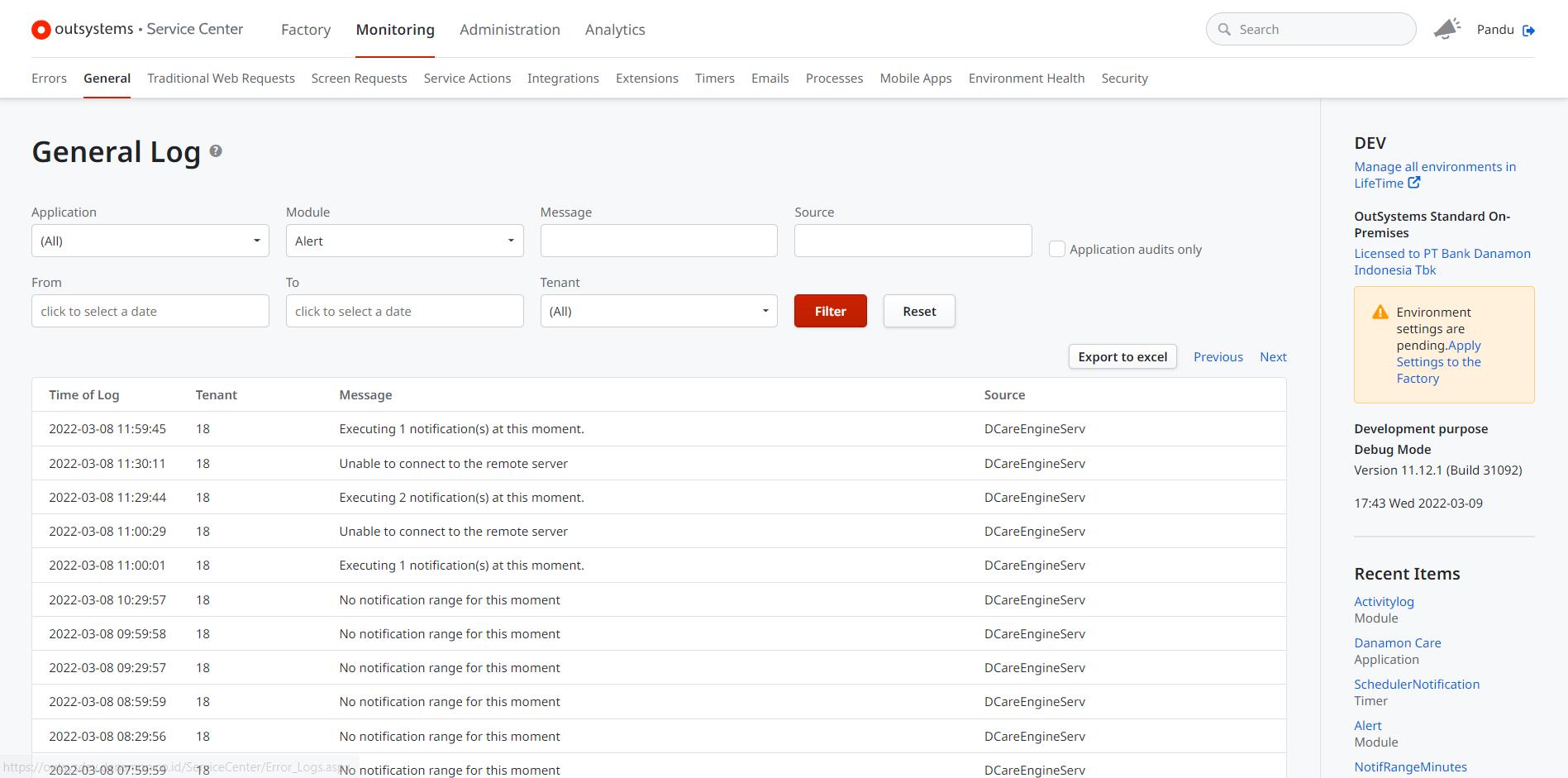
Jika, value @SurveyId = 0 maka jalankan query untuk get user. Sebaliknya, jalankan query get user berdasarkan user sudah melakukan survey atau belum.



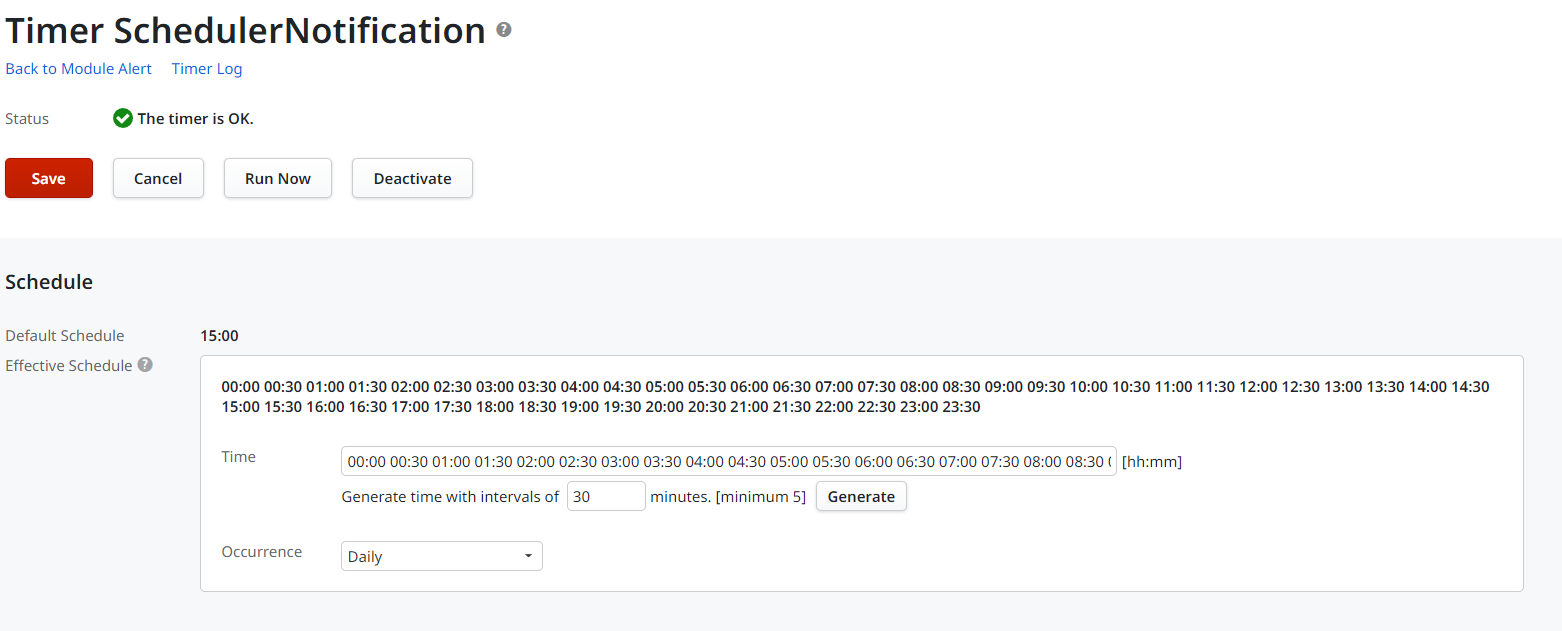
Untuk spesifik group user, tidak ada perubahan.

Untuk memantau jalannya notification developer bisa masuk ke Service Center. Setelah itu ke halaman Monitoring > Tab General. Filter berdasarkan Application, Module atau Source.





Hal yang sangat penting untuk diperhatikan adalah perhatikan time yang digenerate apabila per 30 menit maka site properties NotifRangeMinutes = 29. Jika per 60 menit maka NotifRangeMinutes = 59.

1. Seat Bookings

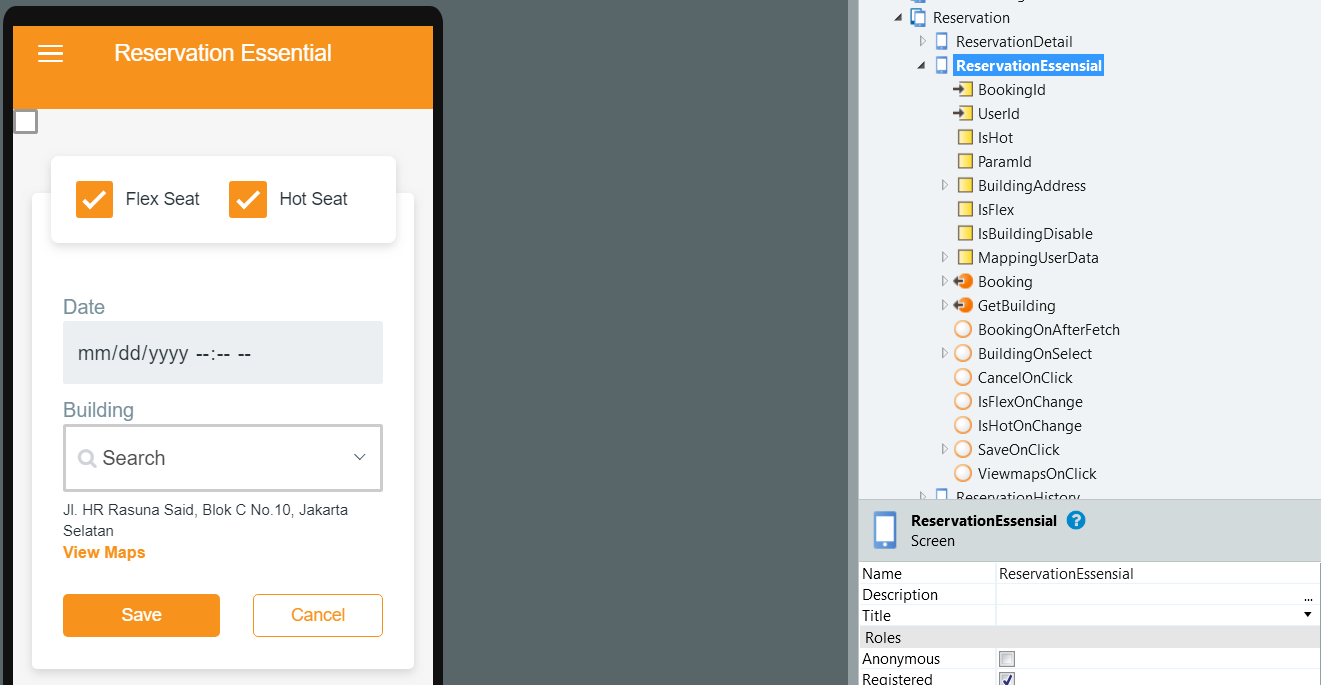
Enhancement Seat Booking terdapat pada aplikasi Clone. Hal ini dilakukan agar tidak berimbas ke aplikasi main. Berikut list clone modules.



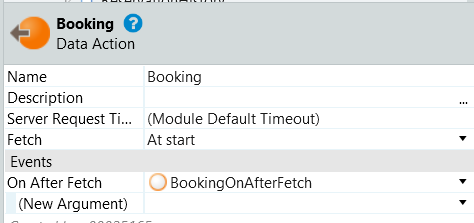
Untuk mekanisme Checker dan Maker pada web backend tidak ada perubahan, menggunakan mekanisme yang sama pada aplikasi main.

Reservation Essential

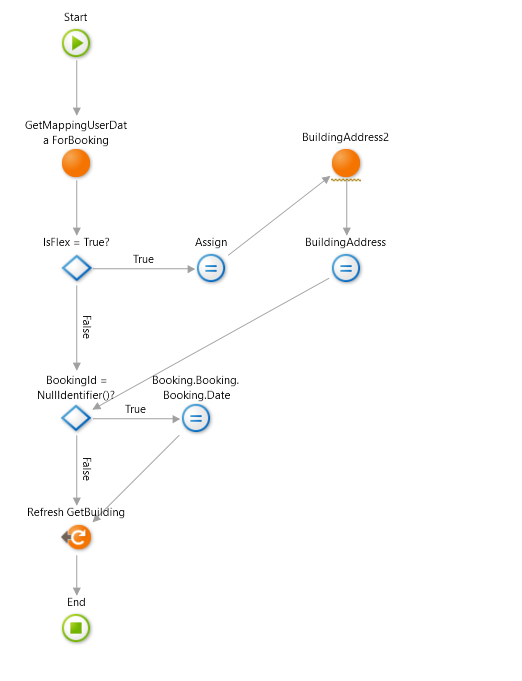
Pada halaman Reservation Essential, terdapat 2 Get Data from other sources yaitu Booking dan GetBuilding.



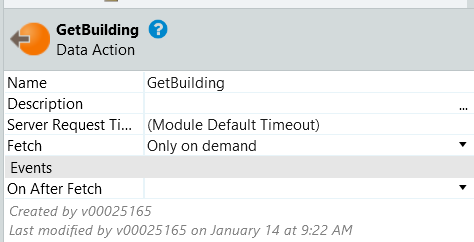
Action Booking merupakan action untuk get data BookingById. Dimana pengaturan propertiesnya adalah Fetch data pada At Start lalu ditambahkan client action OnAfterFetch.



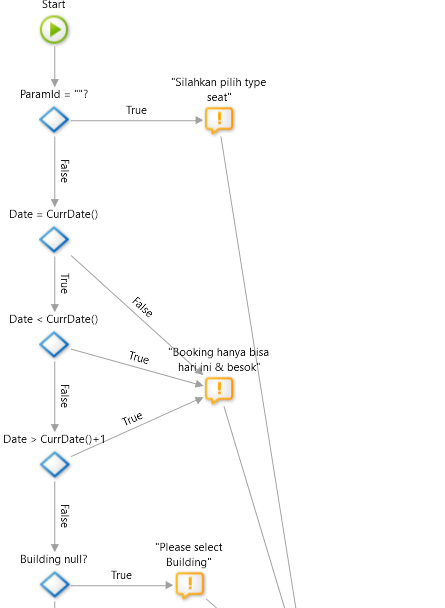
Pada BookingOnAfterFetch, dilakukan validasi pengecekan apakah pengguna merupakan Pegawai Essential atau bukan berdasarkan LOBnya menggunakan action GetMappingUserDataForBooking. Dimana action ini digunakan untuk mendapatkan data pada table master mapping user data. Lalu sistem akan get data building address pengguna tersebut. Setelah itu lakukan refresh data GetBuilding.



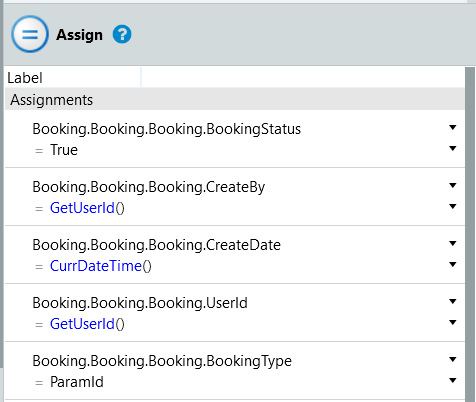
Action GetBuilding digunakan hanya untuk mendapatkan data table master Building.



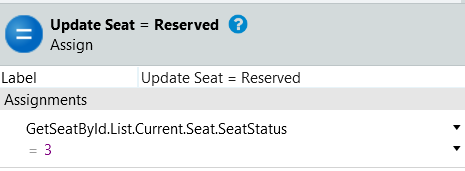
Pada button Save, terdapat beberapa validasi dasar.



Apabila tidak terkena validasi akan ada assignment value berikut. Setelah itu save data Booking.



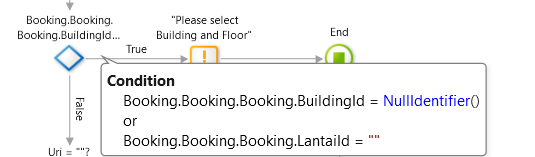
Hal yang perlu diperhatikan adalah beberapa validasi masih hard code. Sebagai contoh pada server action SaveBooking. Untuk update data Seat, langsung dilakukan assign value SeatStatus = 3. Hal ini harus divalidasi pada table master general parameter AMC terlebih dahulu.



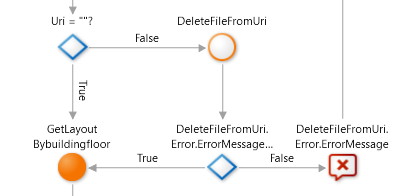
Reservation Non Essential

Pada halaman ReservationNonEssential, terdapat 5 Get Data from other sources yaitu Booking, GetBuilding, GetCapacity, GetSeats dan Lantai. Untuk pengaturan properties Booking sama seperti halaman ReservationEssential sedangkan untuk yang lainnya adalah OnDemand. Begitu juga pada BookingOnAfterFetch. Untuk penyimpanan atau save data Booking pun sama. Yang membedakan adalah pada halaman ini ada beberapa input yang berbeda dan ada view layout ruangan.

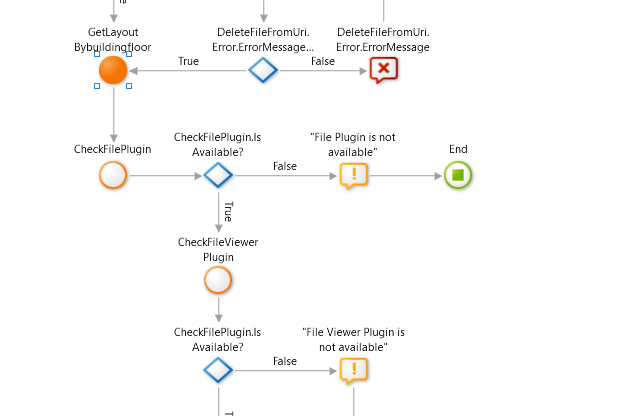
Pada action SeeLayoutOnClick menggunakan plugin CheckFilePlugin dan CheckFileViewerPlugin. Hal pertama yang dilakukan adalah pengecekan apakah pengguna sudah memilih gedung atau lantai.



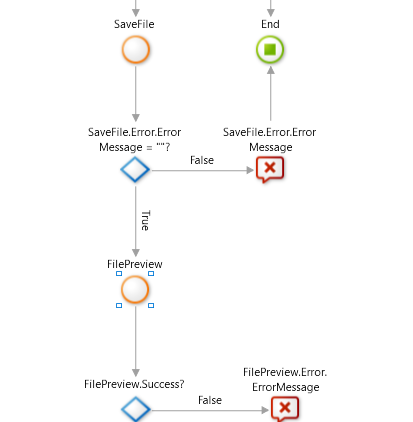
Setelah itu cek apakah variable Uri = “”. Jika tidak, segera clear data file menggunakan action plugin DeleteFileFromUri.



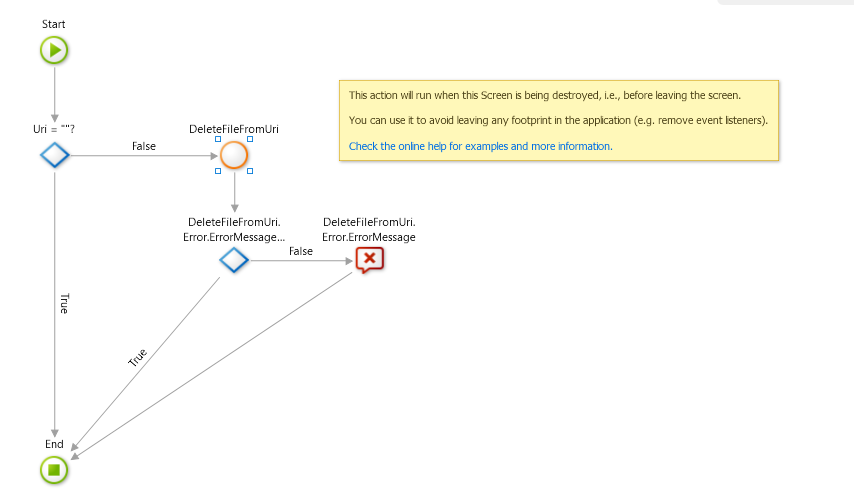
Setelah itu gunakan action GetLayoutByBuildingFloor untuk mendapatkan value File (Binary) dan Filename (Text). Lakukan pengecekan plugin CheckFilePlugin dan CheckFileViewerPlugin.



Apalah sukses melakukan pemeriksaan plugin, gunakan client action FilePlugin > SaveFile untuk menyimpan disebuah spesifik folder. Untuk menampilkan file, gunakan client action FileViewerPlugin > FilePreview. Sistem akan otomatis membuka directory dimana file tersimpan.

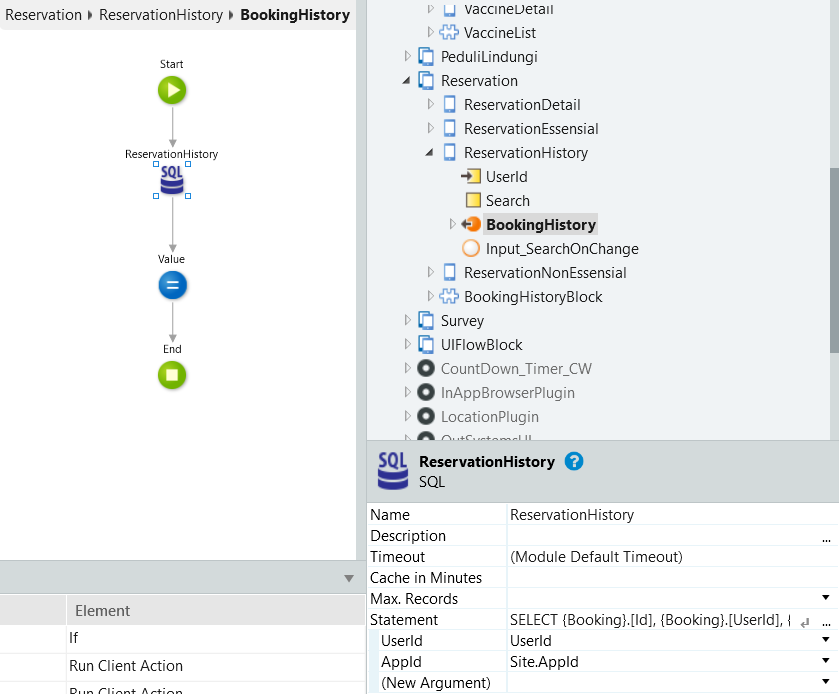


Sebagai catatan, agar terhindar error dari plugin. Uri harus selalu dikosongkan di directory pada saat pindah halaman. Gunakan Screen Action OnDestroy untuk menghapus file dari Uri.



Reservation History

Pada halaman ini hanya menampilan list data Reservasi pengguna. Terdapat 1 input parameter UserId yang digunakan untuk filter data table Reservation.



Reservation Detail

Pada halaman terdapat 3 fitur utama, yaitu Check In, Check Out dan Cancel Reservation.

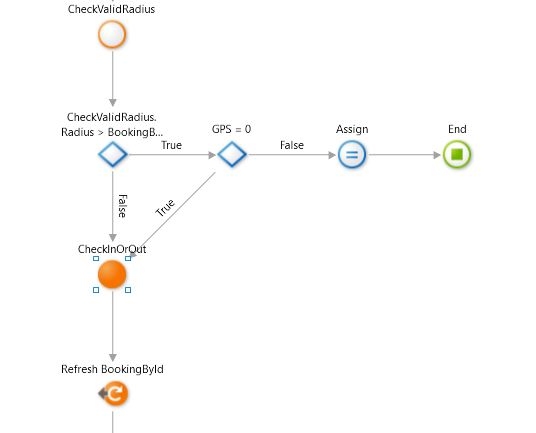
Check In dan Check Out

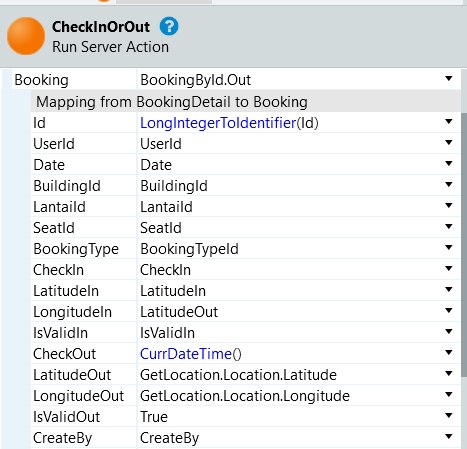
Pada masing-masing fungsi ini terdapat 3 kondisi Check In atau Out.

1. Valid Check In/Out.
2. Invalid Check In/Out karena GPS tidak menyala.
3. Invalid Check In/Out karena di luar Radius.

Valid Check In/Out

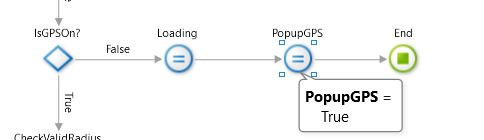
Kondisi Check In/Out valid apabila tidak terkena validasi-validasi pada action CheckIn/OutOnClick. Maka kolom-kolom table Booking harus diisi semua.



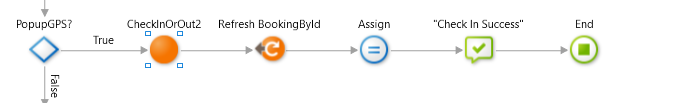


Invalid Check In/Out karena GPS Tidak Menyala

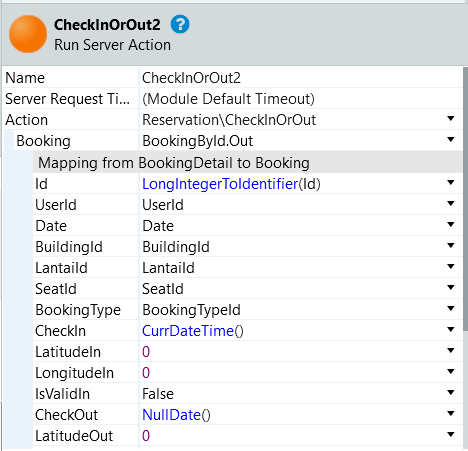
Kondisi Check In/Out Invalid karena GPS Tidak Menyala apabila terkena validasi IsGPSOn pada OnIntialize. Lalu assign variable PopupGPS = True agar pop up konfirmasi Check In/Out GPS muncul.



Apabila pengguna melakukan Check In/Out saat GPS Off, maka sistem akan menjalan flow berikut.

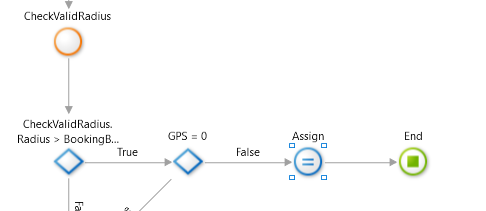


Dimana kolom-kolom terisi kecuali kolom LatitudeIn, LongitudeIn, IsValidIn akan kosong. Untuk Check Out maka LatitudeOut, LongitudeOut dan IsValidOut akan kosong.

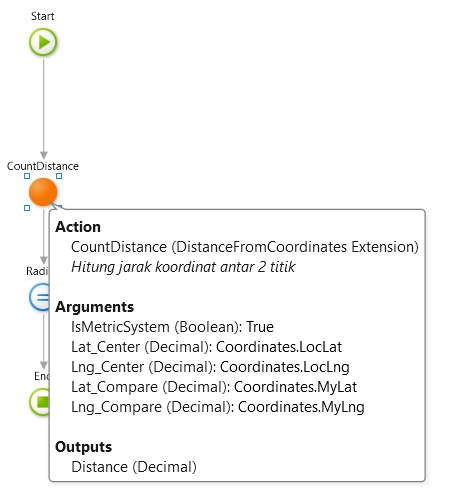


Invalid Check In/Out karena di luar Radius

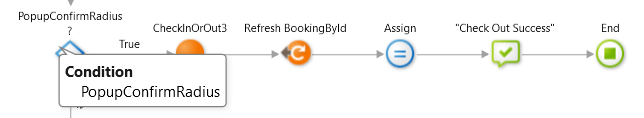
Kondisi Check In/Out karena di luar Radius apabila terkena validasi CheckValidRadius. Jika terkena assign variable PopupConfirmRadius = True agar pop up konfirmasi Check In/Out Radius muncul.



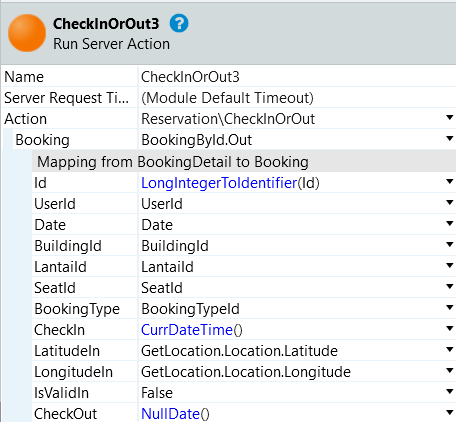
Pada client action CheckValidRadius ada perhitungan jarak menggunakan plugin DistanceFromCoordinates > CountDistance



Apabila pengguna melakukan Check In/Out saat di luar radiues, maka sistem akan menjalan flow berikut.



Dimana kolom-kolom terisi kecuali kolom IsValidIn = False. Untuk Check Out maka IsValidOut = False.



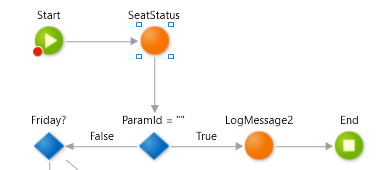
Sebagai catatan, server action CheckInOrOut selain mengupdate status Booking, action ini juga mengupdate status table master Seat

Cancel Reservation

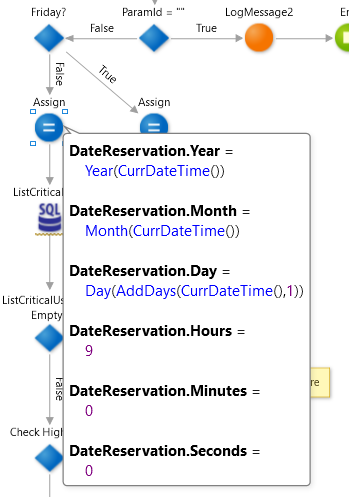
Action ini hanya mengupdate status Booking dan status table master Seat.

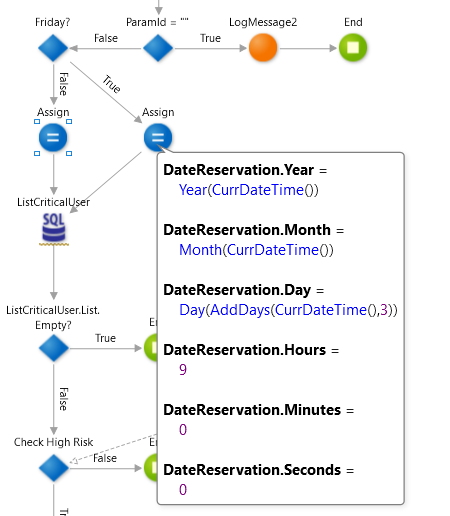
Scheduler Booking User Critical

Terdapat scheduler untuk booking otomatis bagi user critical. Scheduler ini akan melakukan pengecekan SeatStatus terlebih dahulu.

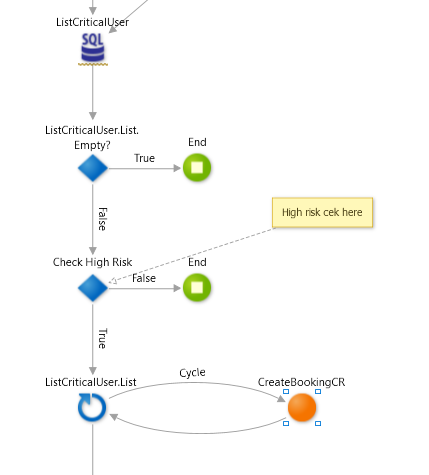


Apabila berhasil get paramid seat status, assign date reservation. Pada booking user critical, terdapat pengecekan apakah hari timer berjalan adalah hari jumat. Apabila hari jumat date reservation akan diisi untuk hari senin esoknya. Jika bukan, maka date reservasi akan diisi untuk hari esoknya.





Jika sudah di assign, hal selanjutnya ada get data critical user menggunakan SP SP\_CrticalUser. Selanjutnya adalah create data booking bagi user critical lalu update seat status.



Catatan: pengecekan user High Risk belum dilakukan.