

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Implementasi *Robotic Process Automation* untuk PT Asuransi Tugu Pratama Indonesia dilakukan oleh dua orang dalam satu project. Tugas yang dilakukan adalah untuk mengimplementasikan robot RPA pada proses operasional *payment request* dan kwitansi. Dalam proses kerja magang seorang *intern* akan dibimbing langsung oleh Bu Therris sebagai *Technical Consultant/Developer*. Selain membimbing, Bu Therris juga akan memberikan informasi penting terkait dengan *payment request* dan kwitansi. Berikut tabel tugas yang akan dilakukan untuk mengimplementasi *Robotic Process Automation* pada PT Asuransi Tugu Pratama Indonesia.

Table 3.1 Tugas Implementasi

| Nama Proses | Human Power |
|-----------------|--|
| Kwitansi | 1 Intern, 1 Technical Consultant Developer |
| Payment Request | 1 Intern, 1 Technical Consultant Developer |

Bentuk koordinasi yang dilakukan adalah dengan menggunakan Microsoft Teams dan Rocket Chat sebagai alat untuk berkomunikasi secara *virtual* atau *work from home* (WFH). Microsoft Share Point dan Jira Confluence digunakan untuk menyimpan informasi seperti Word, Excel, dan Power Point. Microsoft Teams juga dapat digunakan untuk menentukan waktu *meeting* antara *team internal* departemen

maupun diluar departemen. Namun dalam beberapa kesempatan, *developer technical* menentukan waktu untuk *meet-up* tatap muka di CoHive, The Breeze. Pertemuan ini biasanya dibuat untuk membahas *roadblock* dalam proses pengerjaan RPA.

3.2 Tugas yang dilakukan



Gambar 3.1 Gambaran umum implementasi robot RPA

Dalam gambar 3.1, menjelaskan tentang gambaran umum dari proses implementasi *robot* RPA. Proses pembuatan dimulai dari *process design document* (PDD) dan *solution design document* (SDD), implementasi, dan *testing*. Tugas yang akan dilakukan adalah membuat proses automasi kwitansi dan *payment request*. Sebelum pengerjaan dimulai, tahap pertama yang dilakukan adalah menganalisis kebutuhan dasar sebelum mengimplementasi *robotic process automation* (RPA). Tahap tersebut dilakukan dengan menulis dokumen *process design document* (PDD) dan *solution design document* (SDD). Dalam situs www.uipath.com, *process design document* (PDD) adalah sebuah *outline* dari *business proses* yang diberikan UiPath untuk menjelaskan urutan langkah-langkah yang akan dilakukan. Dokumen ini juga merupakan dasar bagi *developer*, untuk memberikan detail yang diperlukan untuk menerapkan *robotic process automation* (RPA).

Tahap kedua adalah tahap pengerjaan proses, dimulai dari proses kwitansi dan kemudian *payment request*. Di dalam proses kwitansi terdapat dua proses besar yaitu proses kwitansi *single*, dan proses kwitansi *multiple*. Hasil dari proses

kwitansi berguna untuk membuat kwitansi fisik maupun kwitansi digital. Sedangkan dalam proses *payment request* terdapat dua proses besar yaitu proses otomatis dan proses manual. Hasil dari proses *payment request* ini berguna untuk menggabungkan antara *file payment request number* dan juga *voucher number*. Pembuatan *robot* RPA menggunakan *platform* UiPath dengan Bahasa pemrograman DotNet dan Linq.

Tahap terakhir yang dilakukan adalah melakukan proses *deployment* dan *test case*. Dalam proses *deployment*, hasil rancangan akhir *robot* akan di *publish* ke dalam *virtual machine* (VM) yang sudah disediakan oleh departemen IT PT Asuransi Tugu Pratama Indonesia. Dan didalam proses *test case* akan dibandingkan dengan hasil dari *robot* RPA. Pengerjaan RPA dilakukan selama empat bulan dimulai dari bulan November hingga Februari terdapat pada tabel 3.2.

Table 3.2 Tabel kegiatan per minggu.

| Minggu ke- | Nama Kegiatan |
|------------|--|
| 1 | Overview high level business model |
| 2 | Proses Detailing business model dan PDD |
| 3 | Proses Pengerjaan PDD |
| 4 | Proses Pengerjaan SDD |
| 5 | Pengerjaan Kwitansi Multiple, Migrate to VM, Replace TTD |
| 6 | Pengerjaan Kwitansi Single, Download Excel, Copy File, Make Report |
| 7 | Testing kwitansi single dan multiple UAT |
| 8 | Debuging Kwitansi single dan multiple |
| 9 | Debugging Kwitansi, dan testing dengan client |

Table 3.3 Tabel kegiatan per minggu. (lanjutan)

| Minggu ke- | Nama Kegiatan |
|------------|---|
| 10 | Debugging Kwitansi |
| 11 | Pengerjaan Payment Request, Manual dan Otomatis |
| 12 | Debuging Payment Request |
| 13 | Pengerjaan Payment Request, Merge PDF |
| 14 | Pengerjaan Payment Request, pindah ke VM |
| 15 | Testing Payment Request, dan Debugging UAT |
| 16 | Debugging Payment Request |

3.3 Uraian Kerja Magang

Dalam minggu pertama dan kedua, pekerjaan yang dilakukan adalah melakukan proses *overview* dan *detailing* terhadap model bisnis PT Asuransi Tugu Pratama Indonesia. Dan juga memahami penggunaan Bahasa DotNet, dan Linq. Proses *overview* dilakukan secara individu sambil menunggu pembuatan infrastruktur RPA, dan jadwal meeting.

Di minggu ke tiga dan ke empat, pembuatan PDD dan SDD dilakukan bersama dengan *client* menggunakan Microsoft Teams sebagai alat pendukung untuk *meeting virtual*. Dalam hasil *meeting* pengembangan RPA dilakukan dengan cara menggunakan *robot attended* yang dimana masih memerlukan manusia untuk melakukan *trigger* agar *robot* berfungsi. Pembuatan PDD dan SDD menggunakan *platform* Jira Confluence sebagai tempat untuk berkolaborasi antar *developer*. Jira Confluence adalah sebuah *web application* yang dapat digunakan untuk berkolaborasi, dan mengorganisir pekerjaan dalam satu tempat, dimana sebuah

team dapat berbagi pengetahuan dan berkolaborasi (Atlassian, 2021). Salah satu keunggulan Jira Confluence adalah dapat mengintegrasikan *platform* lain ke dalam sebuah *workspace* Jira. Setelah pembuatan dokumen PDD dan SDD selesai, dilakukan proses persetujuan antar kedua belah pihak.

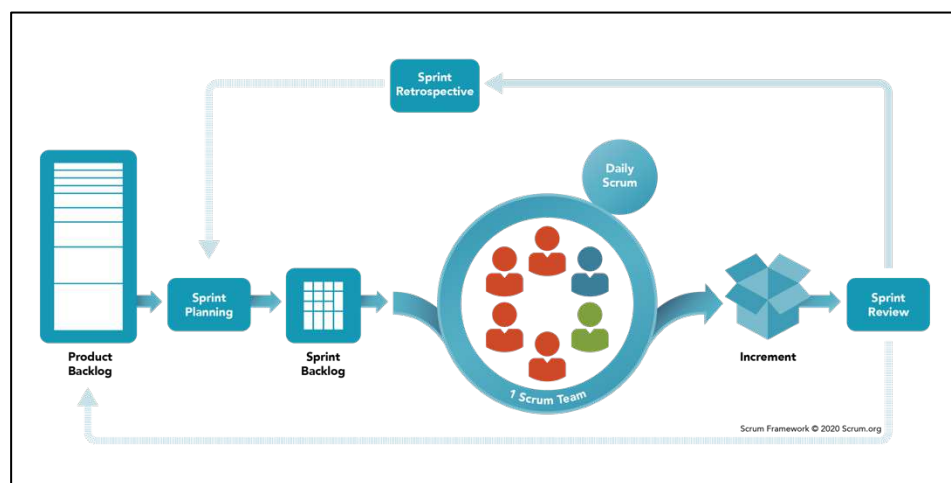
Di minggu ke lima hingga terakhir, pengerjaan proses kwitansi, dan *payment request* dilakukan dengan cara memakai *virtual machine* (VM) yang telah disediakan oleh PT Asuransi Tugu Pratama Indonesia. Pemakaian VM harus disertai dengan penggunaan *virtual private network* (VPN) agar dapat mengakses jaringan *internal* Asuransi Tugu seperti *shared folder* dan sistem care/informasi Tugu. Terdapat tiga VM yang disediakan oleh PT Asuransi Tugu Pratama Indonesia yaitu dua VM *developer*, dan satu VM *production*. Dua VM *developer* tersebut digunakan oleh *technical developer*, dan *intern*, untuk mempercepat pembuatan *robot* RPA. Setelah *robot* selesai dibuat, hasil pengerjaan di-*publish* lalu di-*deploy* ke dalam VM *production*.

Aplikasi yang dibutuhkan untuk membuat robot RPA ini adalah UiPath, PDFSam, Google Chrome, MS Word, dan MS Excel. UiPath digunakan sebagai platform untuk mengembangkan robot RPA. PDFSam digunakan untuk melakukan proses merge PDF overbooking dan *payment request*. Google Chrome digunakan untuk membuka sistem informasi tugu. Kemudian MS Word dan Excel digunakan untuk melakukan proses operasional antarmuka robot.

Kemudian pada minggu ke tujuh dan lima belas dilakukan proses pengujian dengan uji *test case*. Metode *test case* dipilih untuk menguji kecacatan dalam pembuatan *robot* RPA. Pengujian ini bertujuan untuk membandingkan hasil dari implementasi robot RPA dengan *test case*.

3.3.1 Proses Pelaksanaan

Proses pelaksanaan dilakukan dengan metode scrum. *Scrum* merupakan salah satu model dari Agile Methodology. Menurut Schwaber, “*scrum* adalah sebuah *framework* ringan yang dapat membantu orang, *team*, atau organisasi untuk menghasilkan *value* melalui solusi adaptif untuk masalah kompleks.” (Schwaber & Sutherland, 2020). Metode ini digunakan karena sifatnya yang adaptif terhadap perubahan sehingga membuat pembuatan *robot* RPA lebih *robust*.



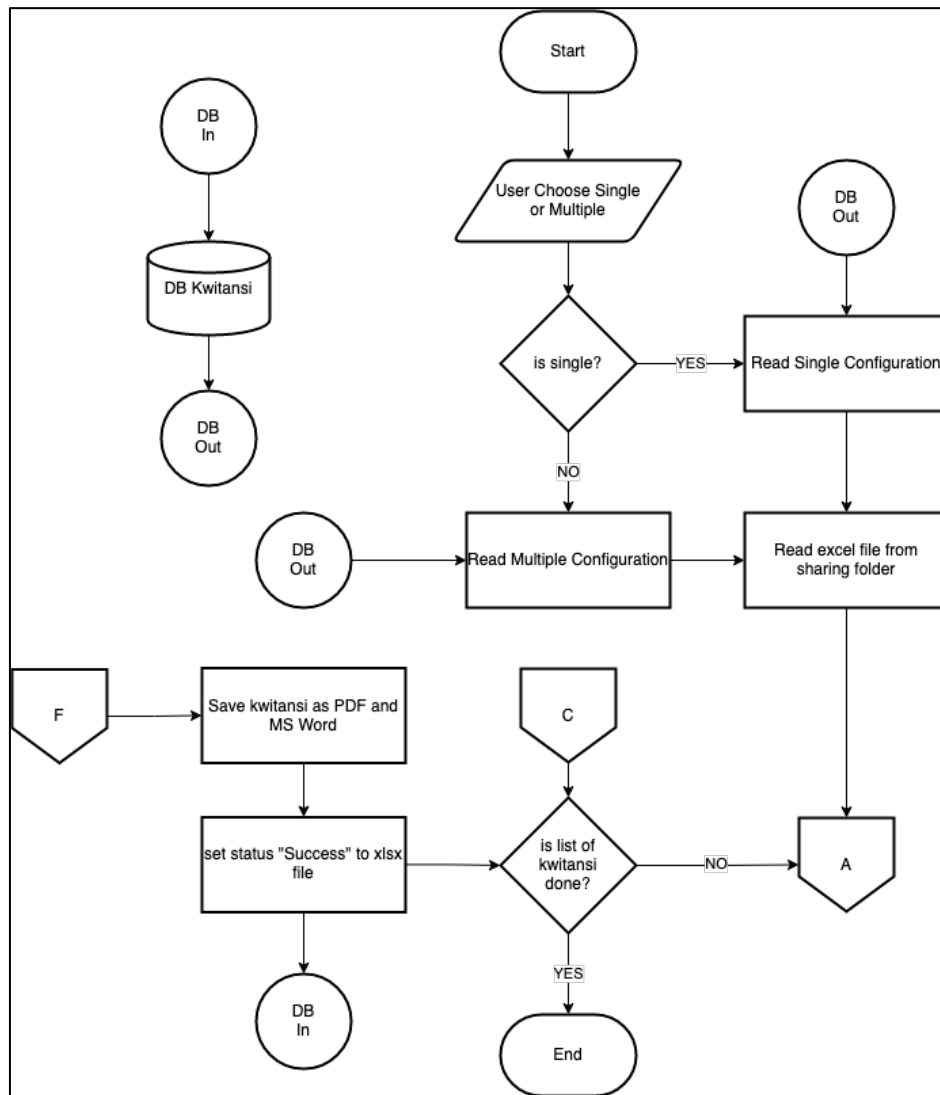
Gambar 3.2 *Scrum Methodology*.
(Sumber: www.scrum.org, 2021)

Secara garis besar *scrum team* terbagi menjadi *scrum master*, *product owner*, dan *developers*. Dalam gambar 3.2, menjelaskan proses *scrum*, yang dimana *product owner* menyusun *product backlog*, kemudian diberikan kepada *developers* untuk mengatur *sprint planning*. Setiap satu *sprint* biasanya terbentuk dari satu hingga empat minggu, kemudian dalam sehari *scrum team* harus membuat *daily scrum* maksimal 15 menit untuk memeriksa dan menyesuaikan kemajuan progress menuju *sprint goal*. *Scrum master* bertujuan untuk memberikan *coaching* terhadap *scrum team* agar dapat berfokus dengan *scrum theory*. Di akhir *sprint*, *scrum team* akan mengundang *client* atau *stakeholders* untuk melakukan *sprint review*.

Kemudian *scrum team* akan menjalankan *sprint retrospective* untuk mengevaluasi dan mengembangkan cara kerja. (Schwaber & Sutherland, 2020).

A. Flowchart

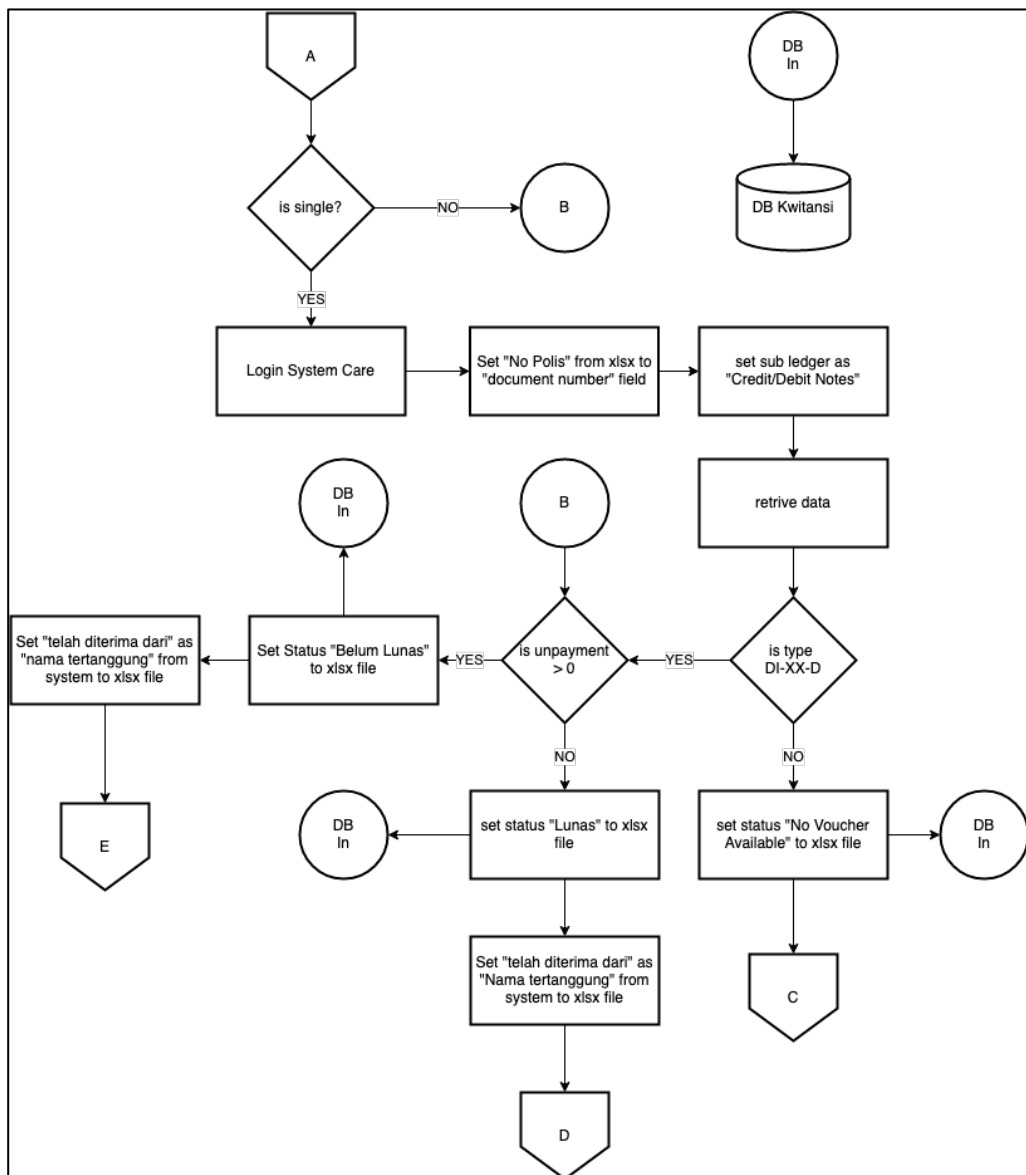
A.1 Flowchart Kwitansi



Gambar 3.3 Flowchart Kwitansi 1

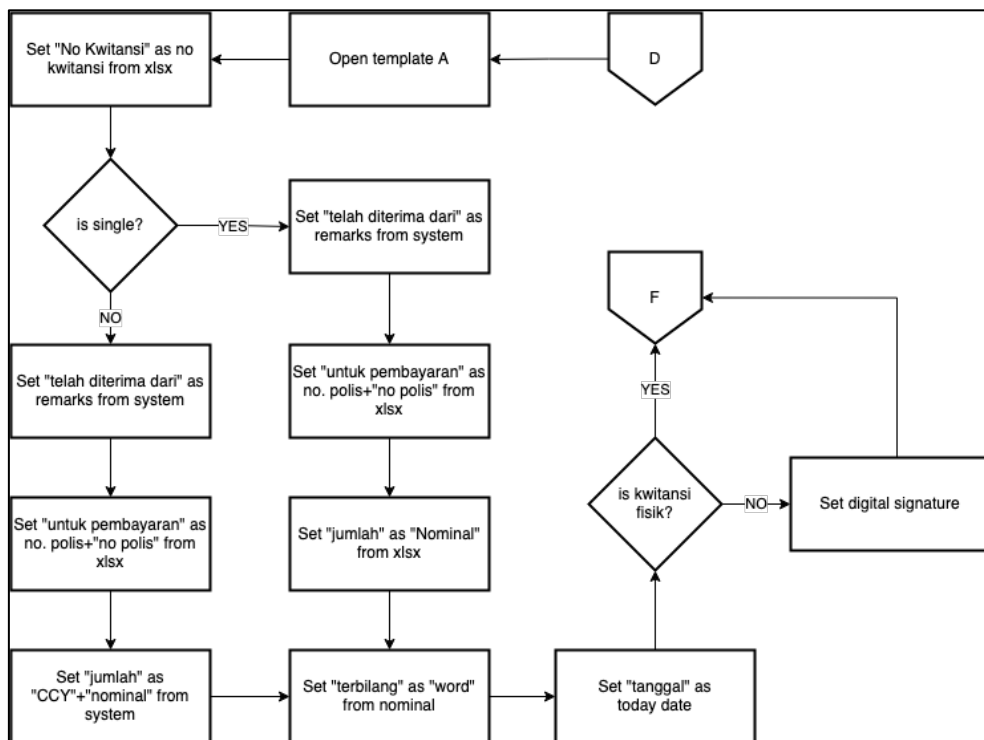
Dalam gambar 3.3, proses kwitansi berjalan saat user memilih tipe antara *single* kwitansi atau *multiple* kwitansi. Setelah user memilih tipe, *robot* RPA akan membaca konfigurasi *robot*, yang berisi *path* dari sebuah *folder* untuk di-*execute*, dan robot juga akan membaca *shared folder* yang akan diproses menjadi kwitansi

digital maupun kwitansi fisik. Kemudian *robot* akan lanjut ke *off page reference A*. Berbeda dari proses *off page reference F*, *robot* akan melakukan *save* ke dalam PDF dan juga MS Word. Setelah itu *robot* akan mengecek apakah list dari kwitansi sudah dibaca semua atau tidak. Jika list kwitansi masih ada, maka *robot* akan lanjut ke *off page reference A* lagi. Demikian juga dari *off page reference C*. Namun, jika list kwitansi sudah selesai, maka *robot* akan berhenti.



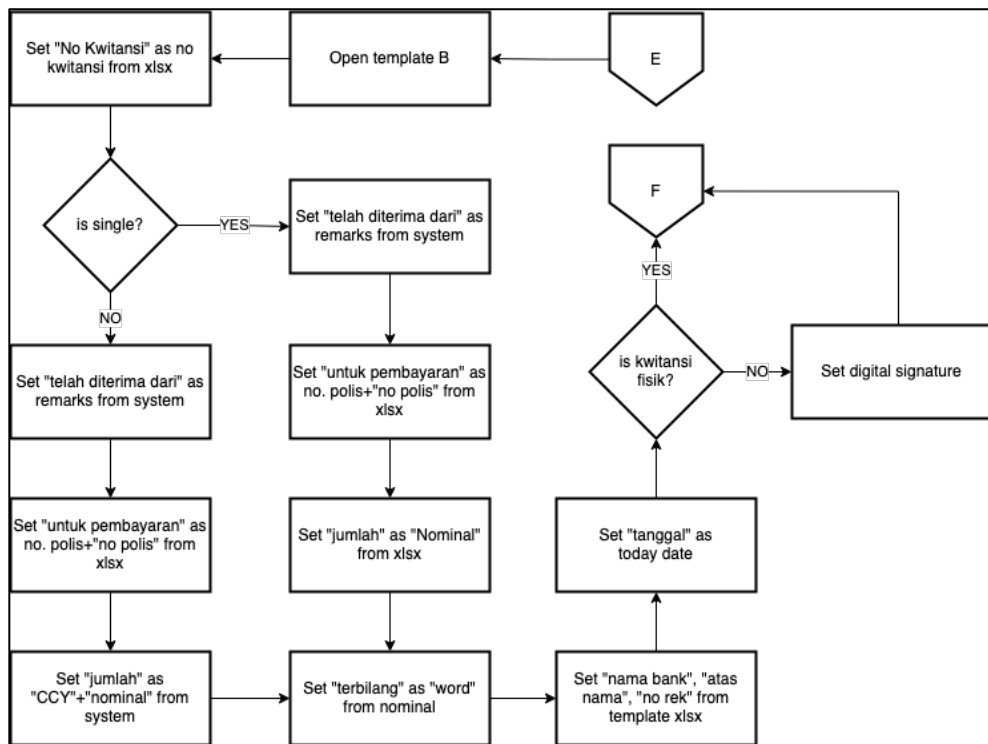
Gambar 3.4 *Flowchart Kwitansi 2*

Dalam gambar 3.4, proses *off page reference* A, *robot* akan mengecek tipe yang di-*input* oleh *user* apakah *single* atau *multiple*. Jika tipenya *single*, maka *robot* akan melakukan proses tambahan yaitu dengan *login* ke sistem *care*, mengeset nomor polis dari *file excel* yang sudah disediakan pada saat diawal konfigurasi, dan mengeset *subledger* menjadi “*Credit/Debit Notes*”. Setelah itu, *robot* akan mengecek tipe *retrieve* data apakah sesuai dengan *overbooking voucher/DI-02-D* atau tidak. Jika tidak sesuai *robot* akan menulis status “*no voucher available*” dan mengecek list kwitansi yang lain. Namun jika tipenya *multiple*, maka *robot* akan langsung mengecek *unpayment*. jika *unpayment* lebih besar dari nol maka *robot* akan melakukan proses belum lunas, sebaliknya jika *unpayment* lebih kecil atau sama dengan nol maka *robot* akan melakukan proses lunas.



Gambar 3.5 Flowchart Kwitansi 3

Dalam gambar 3.5, proses *off page reference* D, atau proses kwitansi lunas, robot akan membuka template A dan kemudian akan mengeset “nomor kwitansi”, “telah diterima dari”, “untuk pembayaran”, “jumlah”, “terbilang”, dan “tanggal” hari ini. Khusus untuk terbilang, penggunaan nominal harus menggunakan dengan kalimat. Setelah itu *robot* akan mengecek kwitansi fisik atau digital. Jika fisik, *robot* akan lanjut ke *off page reference* F. Jika digital, *robot* akan membaca tanda tangan digital lalu memasukkannya ke dalam MS Word, dan lanjut ke *off page reference* F pada gambar 3.3.

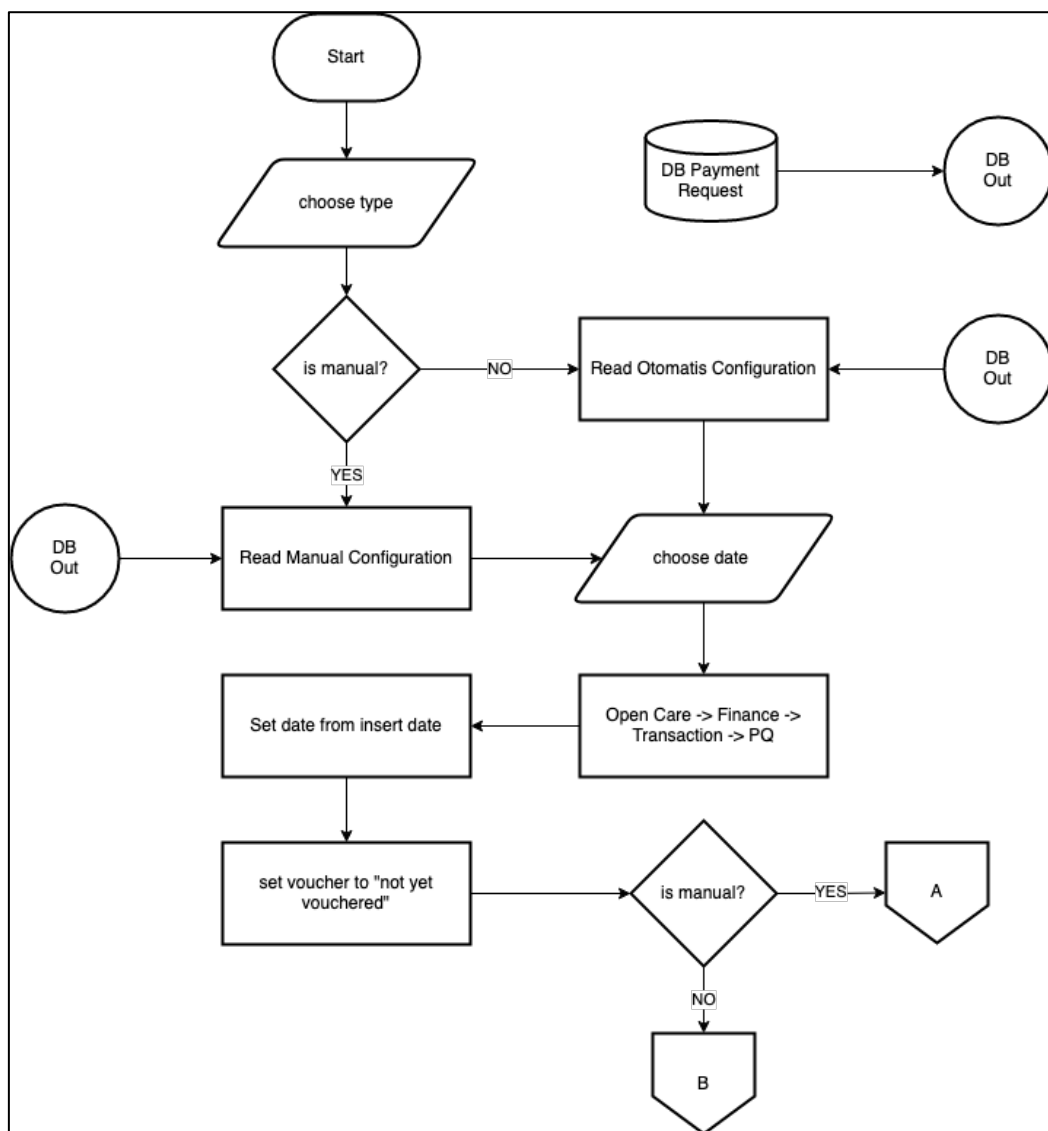


Gambar 3.6 *Flowchart* Kwitansi 4

Dalam gambar 3.6, proses *off page reference* E, atau proses kwitansi belum lunas, *robot* akan membuka *template* B dan kemudian akan mengeset “nomor kwitansi”, “telah diterima dari”, “untuk pembayaran”, “jumlah”, “terbilang”, “nama bank”, “atas nama”, “no rek”, dan “tanggal” hari ini. Khusus untuk terbilang, penggunaan nominal harus menggunakan dengan kalimat. Setelah itu *robot* akan

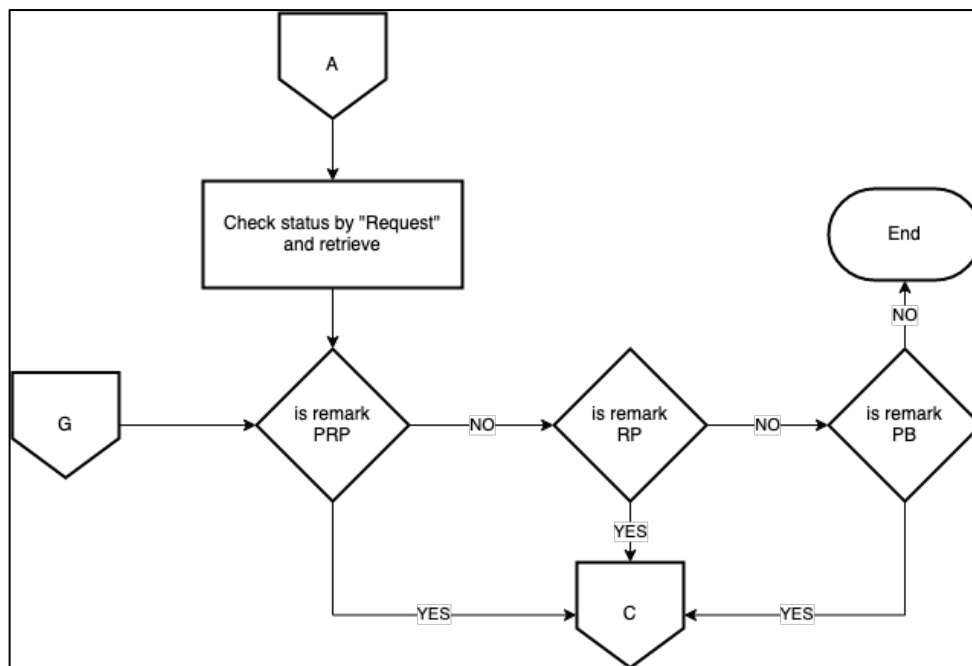
mengecek kwitansi fisik atau digital. Jika fisik, *robot* akan lanjut ke *off page reference F*. Jika digital, *robot* akan membaca tanda tangan digital lalu memasukkannya ke dalam MS Word, dan lanjut ke *off page reference F* pada gambar 3.3.

A.2 Flowchart Payment Request

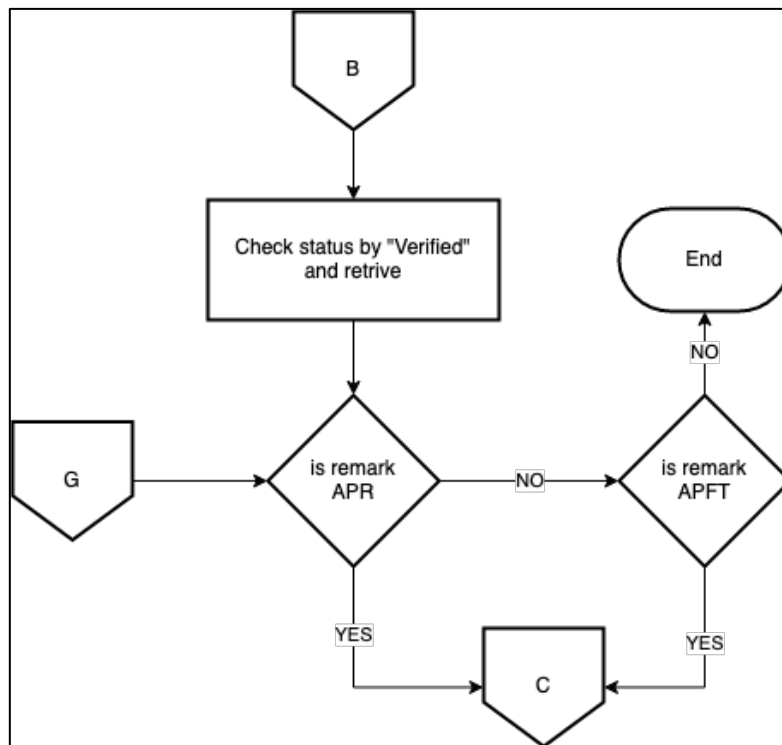


Gambar 3.7 Flowchart Payment Request 1

Dalam gambar 3.7, proses *payment request* berjalan saat user memilih tipe antara manual atau otomatis. Setelah user memilih tipe, *robot* RPA akan membaca konfigurasi *robot*, yang berisi *path*, dan konfigurasi *robot* agar dapat di-*execute*. Kemudian *robot* akan menampilkan *dialog box* untuk memasukkan *start date* dan *end date*, dalam pilihan ini *user* dapat memasukkan *date* dengan tipe dd/mm/yyyy. Setelah memasukkan *date*, *robot* akan membuka sistem informasi tugu/care, untuk mengambil *file payment request*. Lalu, *robot* akan mengecek tipe, jika tipe manual *robot* akan lanjut ke *off page reference* A pada gambar 3.8, jika otomatis, *robot* akan lanjut ke *off page reference* B pada gambar 3.9.

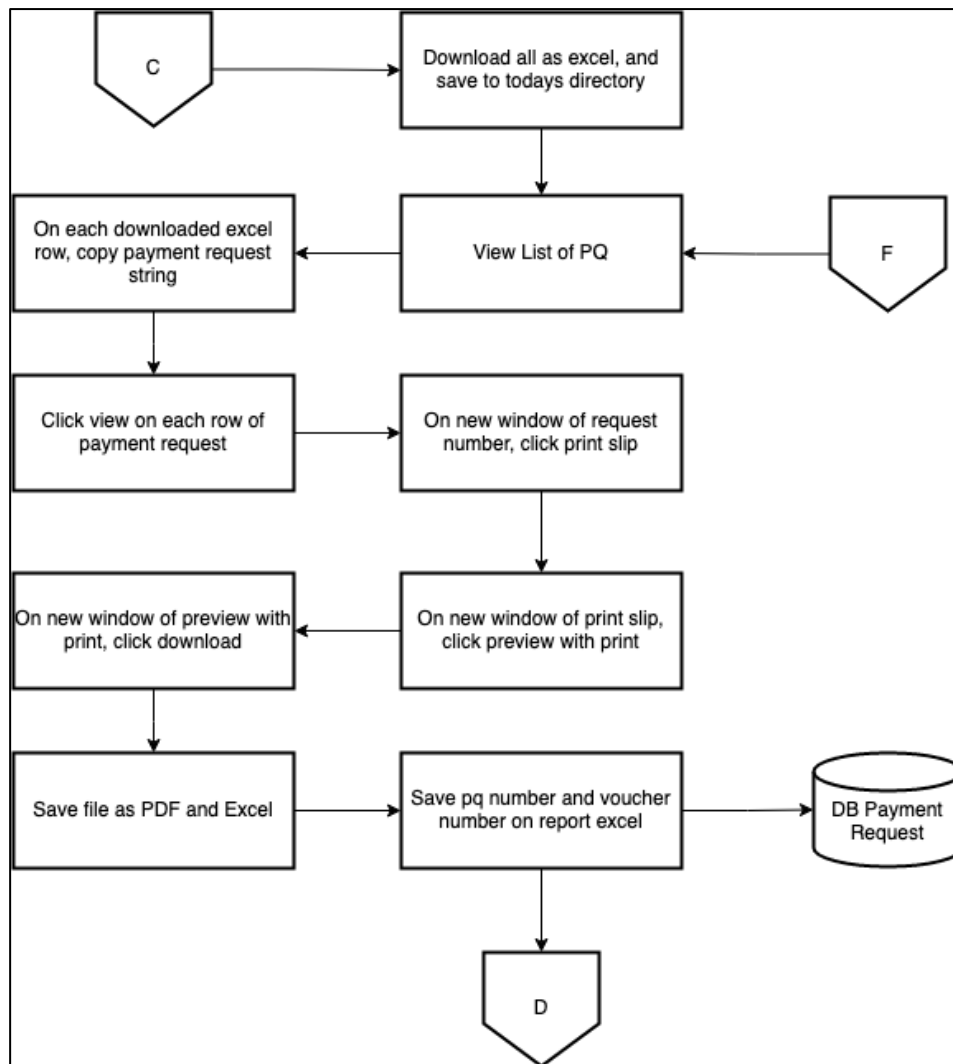


Gambar 3.8 *Flowchart Payment Request 2*



Gambar 3.9 Flowchart Payment Request 3

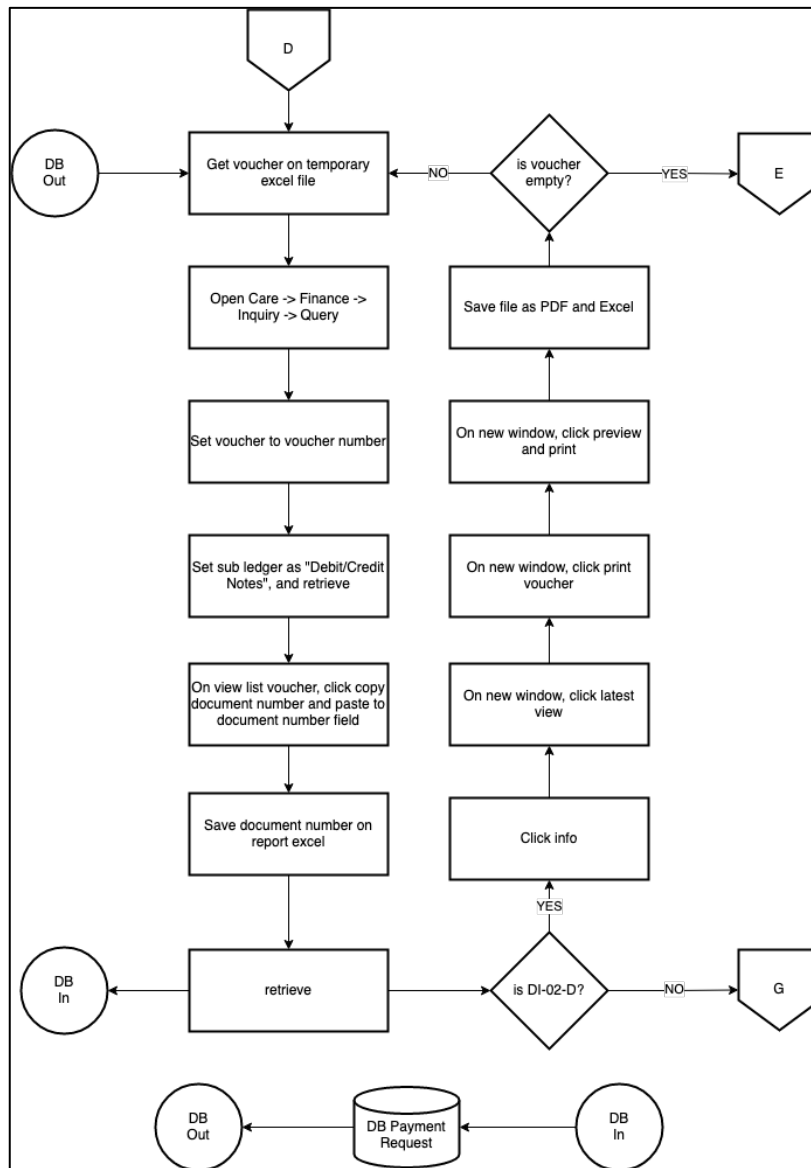
Dalam gambar 3.8, pada *off page reference A*, robot akan mengecek setiap *remarks* berdasarkan tipe manual yang meliputi *payment request process*, *refund premi*, dan *pemindahbukuan*, kemudian robot akan melanjutkan proses pada *off page reference C*. Dan dalam gambar 3.9, dari *off page reference B*, robot akan mengecek setiap *remarks* berdasarkan tipe otomatis yang meliputi *auto payment request*, dan *auto payment from tfriends*, kemudian robot akan melanjutkannya pada *off page reference C*. Ketika setiap *remarks* sudah dicek, maka robot akan berhenti. Untuk *off page reference G*, terjadi Ketika setiap *remarks* diproses dan dilakukan pengecekan ulang.



Gambar 3.10 Flowchart Payment Request 4

Dalam gambar 3.10, *robot* akan melanjutkan proses dari *off page reference* C untuk men-*download* semua data yang telah di-*retrieve* sebelumnya. Lalu, setelah men-*download file* *robot* akan membaca tiap list *payment request*. Kemudian *robot* akan melakukan *print slip*, untuk men-*download* dan menyimpannya sebagai PDF dan Excel, lalu *robot* juga akan mengambil *voucher number* dari hasil *downloaded file* sebelumnya. Setelah itu *robot* akan melakukan proses penyimpanan *payment*

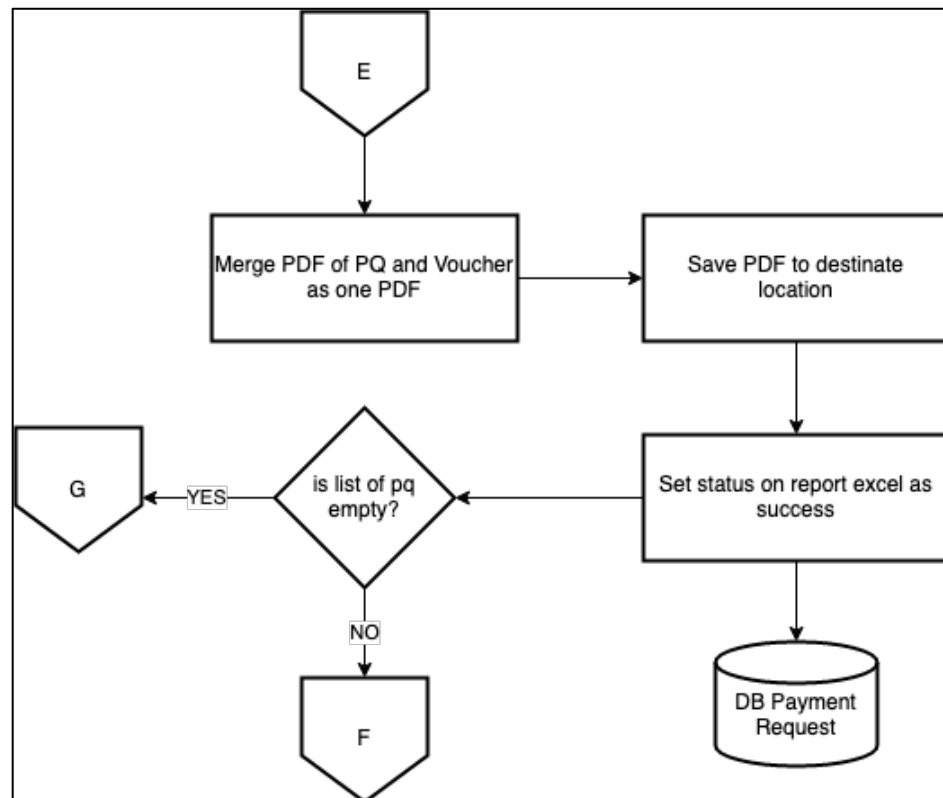
request number dan *voucher number* pada *report excel*. Lalu *robot* akan lanjut ke *off page reference D*.



Gambar 3.11 *Flowchart Payment Request 5*

Dalam gambar 3.11, robot akan melanjutkan proses dari *off page reference D*. Robot akan mengambil *voucher number* dari *report file* yang telah dibuat sebelumnya untuk mencari *file voucher* pada sistem care. Setelah itu *voucher* akan dicari berdasarkan tipe DI-02-D atau *overbooking voucher*, jika tipe sesuai robot akan men-*download* dan menyimpan sebagai PDF dan Excel. Setelah itu robot akan

mengecek *voucher* selanjutnya jika *voucher* masih ada, *robot* akan mengulang proses yang sama, jika tidak *robot* akan lanjut ke *off page reference* E. Namun berbeda dengan tipe yang tidak sesuai dengan *overbooking voucher/DI-02-D*, *robot* akan lanjut ke *off page reference* G.



Gambar 3.12 *Flowchart Payment Request 6*

Dalam gambar 3.12, *robot* akan melanjutkan proses *off page reference* E, yang kemudian melakukan *file merging* antara *file payment request* dan *file voucher*. Setelah itu, *robot* akan menyimpan *file* dan mengeset *status* pada *report file* sebagai *success*. Kemudian *robot* akan mengecek list *payment request* lagi, jika masih ada maka lanjut ke *off page reference* G pada gambar 3.8 dan gambar 3.9, jika tidak proses akan lanjut ke *off page reference* F pada gambar 3.10.

B. Deployment

Proses *deployment* dilakukan ketika pembuatan *robot* RPA sudah selesai. *Robot* tersebut akan di-*publish* menjadi sebuah *package*, yang kemudian dijalankan dengan menggunakan aplikasi *robot assistant*. Aplikasi ini berguna untuk melakukan proses *trigger robot*, bisa dilakukan secara manual maupun secara otomatis. *Package* yang sudah di-*publish* tadi, akan disimpan ke dalam VM *production* untuk melakukan proses *testing* oleh *client*.

C. Hasil Implementasi

1) Kwitansi

KWITANSI – OFFICIAL RECEIPT [REDACTED] /2020

Telah terima dari
Received from
Terbilang
Says
Untuk Pembayaran
Payment of

[REDACTED]

Sebelas Juta Tiga Ratus Sembilan Puluh Ribu Tujuh Ratus Dua Puluh Lima Rupiah

NO POLIS [REDACTED]

Jumlah
Total
IDR 11.390.725.00

Jakarta, 15 February 2021

[REDACTED]

Treasury Group Head

Kwitansi ini dianggap sebagai tanda terima yang sah apabila pembayaran telah diterima di rekening perusahaan.
This receipt is considered as a valid receipt when payment has been received on account of company.

PT. Pratiwi
Jl. Pratiwi, Kav. C, Jakarta 10110
Telp. +62 21 6177 5525 | Fax +62 21 6177 5525

Gambar 3.13 contoh kwitansi hasil *robot*.

2) Payment Request

| | | | |
|---|-------------------------------|------------------------|------------------------|
| PT Asuransi Tugu Pratama Indonesia Tbk | | NO : | ██████████ |
| POSS HO (WismaTugu 1) | | PAYMENT REQUEST | DATE : 14 Januari 2021 |
| PAYMENT TO : | ██████████esting (D13IT59013) | POLICY NO. : | |
| REMARKS : | Payment Request Process | | |
| REFERENCE NO. : | - | TYPE : | TRANSFER |
| N/L ACCOUNT : | | | |

| KODE | KETERANGAN | JUMLAH |
|---------|--|------------------|
| PV-01-P | Payment Voucher - Suspend Premium/Claim | AMOUNT |
| | 14/01/2021 - ██████████ - (Branch: 913103) | IDR 1,268,394.00 |

Gambar 3.14 Contoh hasil *payment request*

| | | | |
|---|---|------------------|------------------|
| PT Asuransi Tugu Pratama Indonesia Tbk | | NO. : | ██████████ |
| POSS PIM | | TGL : | 14 Januari 2021 |
| Ref. Voucher No : | ██████████ | Nominal : | IDR 5,227,120.00 |
| Company Account : | KB-001 - OVERBOOKING & JOURNAL MEMORIAL | Rate : | 1.00 |
| Profile : | ██████████ | Nominal In IDR : | 5,227,120.00 |
| Remarks : | ██████████ | Collection No. : | |

| Voucher No. | Date | DocNo./RefNo. | Ccy | Nominal in OC | Rate | Nominal |
|-------------|------------|------------------------------|-----|---------------|-------|--------------|
| ██████████ | 14/01/2021 | ██████████ tfriends-47108 | IDR | 5,227,120.00 | 1 / 1 | 5,227,120.00 |

Gambar 3.15 Contoh hasil *voucher*

2) Test case

Proses testing dilakukan dengan cara *test case scenario*, yang dimana proses ini dilakukan dengan cara menyamakan hasil *robot* dengan hasil *scenario* yang diharapkan. Berikut hasil *test case* dari pembuatan *robot* RPA:

Table 3.4 Perbandingan pengerjaan manual dan *robot*.

| No | Proses | Periode | Jumlah Data | Waktu Pengerjaan Manual Per Data | Total Waktu Pengerjaan Manual | Waktu Pengerjaan Robot Per Data | Total Waktu Pengerjaan Robot |
|----|----------|---------|-------------|----------------------------------|-------------------------------|---------------------------------|------------------------------|
| 1 | PQ | Harian | 20 | 15 Menit | 5 Jam | 7 Menit | 2 Jam 33 Menit |
| 2 | Kwitansi | Harian | 5 | 15 Menit | 1 Jam 25 Menit | 1 Menit | 5 Menit |

Tabel 3.4, menunjukkan hasil akhir dari setiap 1 pengerjaan *payment request* manual yang menghabiskan 15 menit per data, sedangkan *robot* RPA menghabiskan waktu 7 menit per data, jadi jika jumlah data 20 maka *robot* akan menghabiskan waktu 2 jam 33 menit. Dan, Untuk 1 proses kwitansi manual dapat menghabiskan waktu 15 menit per data, sedangkan *robot* RPA menghasilkan 1 menit per data, jadi jika jumlah data 5 maka *robot* akan menghasilkan waktu 5 menit.

Table 3.5 Contoh *test case scenario payment request* manual

| Process | Test Case | Expected | Actual | Pass/Fail |
|-----------|---|--|---------------------------------------|-----------|
| PQ Manual | Tanggal start date dan end date diisi oleh user | Proses Berlanjut | Proses Berlanjut | Pass |
| | Saat pemilihan tanggal user melakukan input manual tidak sesuai format | Proses Selesai, ada notifikasi Error Input Date, Kembali ke layar untuk input date | Kembali ke layer input date | Pass |
| | Robot akan melakukan set remarks dengan payment request process -- melakukan set Vouchered dengan Not Yet Vouchered --> Robot akan klik Retrieve ---> ada list data ---> Cek status Request | Proses Berlanjut | Proses berlanjut | Pass |
| | Robot melakukan input voucher no pada list xlsx ke Finance, voucher ditemukan -- - DI-XX-D | Proses Selesai, Berlanjut ke proses lain | Proses selesai, lanjut ke proses lain | Pass |

| | | | | |
|--|---|--|----------------|------|
| | Robot melakukan input voucher no pada list xlsx ke Finance, voucher tidak ditemukan | Proses selesai | Proses selesai | Pass |
| | Melakukan Merge Dokumen untuk data yang berpasangan Dokumen PQ dan Dokumen MR | Tidak dilakukan merge document, proses selesai | Proses selesai | Pass |

Table 3.6 Contoh *test case scenario* kwitansi *multiple*

| Process | Test Case | Expected | Actual | Pass/Fail |
|-------------------|--|---|---------------------------------------|-----------|
| Kwitansi Multiple | Robot menerima order pembuatan kwitansi multiple membuka dokumen pada folder tertentu | Proses Selesai (ada notifikasi proses error) | Proses Berlanjut | Pass |
| | Data Excel Sesuai, melakukan pembacaan pada data Excel. Menaruh No Kwitansi pada "No, xxxxxxxx" | Proses Selesai (ada notifikasi no kwitansi tidak ada) | Kembali ke layer input date | Pass |
| | Melakukan pembacaan pada data Excel. Menaruh Nama Tertanggung pada "Telah diterima dari : " | Proses Berlanjut (jika kosong diisi No. atau Proses Selesai) | Proses berlanjut | Pass |
| | Melakukan pembacaan pada data Excel. Menaruh No Polis pada "Untuk Pembayaran : " | Proses Berlanjut (diisi kosong No. atau Proses Selesai) | Proses selesai, lanjut ke proses lain | Pass |
| | Memasukan Jumlah pada data Excel pada "Jumlah" pada template | Proses Berlanjut (jika kosong diisi No. atau Proses Selesai) | Proses selesai | Pass |
| | Menambahkan kolom terbilang yang berisi kalimat dari nominal dari kolom "jumlah" | Proses Berlanjut (diisi Tidak ada data jika kosong / #NA) | Proses Berlanjut | Pass |
| | Melakukan pembacaan Jenis Kwitansi Fisik / E Kwitansi | Proses Berlanjut akan membuat kwitansi dengan format Fisik / E kwitansi | Proses Berlanjut | Pass |
| | Membaca Data Master untuk jabatan Approval Kwitansi | Proses Berlanjut (diisi kosong No. atau Proses Selesai) | Proses Berlanjut | Pass |
| | Proses selesai melakukan save by word dan PDF dan melakukan pemindahan dokumen master awal disertai keterangan status proses di folder sesuai tanggal proses | Proses Selesai | Proses Selesai | Pass |

3.3.2 Kendala yang ditemukan

Terdapat dua kendala yang ditemukan pada saat pengerjaan implementasi RPA. Berikut kendala yang ditemukan:

1. *Roadblock* ketika *selector* UiPath tidak dapat membaca *node UI* pada *dashboard* PDFSam yang sering berubah. Masalah ini mengakibatkan *robot* RPA tidak dapat membaca *selector* sehingga *robot crash*.
2. *Dependency* yang berada di VM *developer* dan di VM *production* tidak dapat di-*install*, karena koneksi *internet* di dalam VM terblokir oleh *firewall*. Masalah ini mengakibatkan UiPath menjalankan proses *retry* yang lama, sehingga memakan waktu pengerjaan.

3.3.3 Solusi atas kendala yang ditemukan

Terdapat dua solusi yang dilakukan pada saat pengerjaan implementasi RPA. Berikut Solusi atas kendala tersebut:

1. Mengubah *setting* PDFSam untuk tidak menampilkan *dashboard*, tetapi hanya *merge document*.
2. *Dependency* pada *local machine* dapat di-*copy and paste* ke dalam *package dependency* VM *developer* dan VM *production*. Karena fitur *copy paste file* dalam *Remote Desktop Connection* (RDC) dapat dilakukan, maka *dependency* tersebut dibuat secara manual.