



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

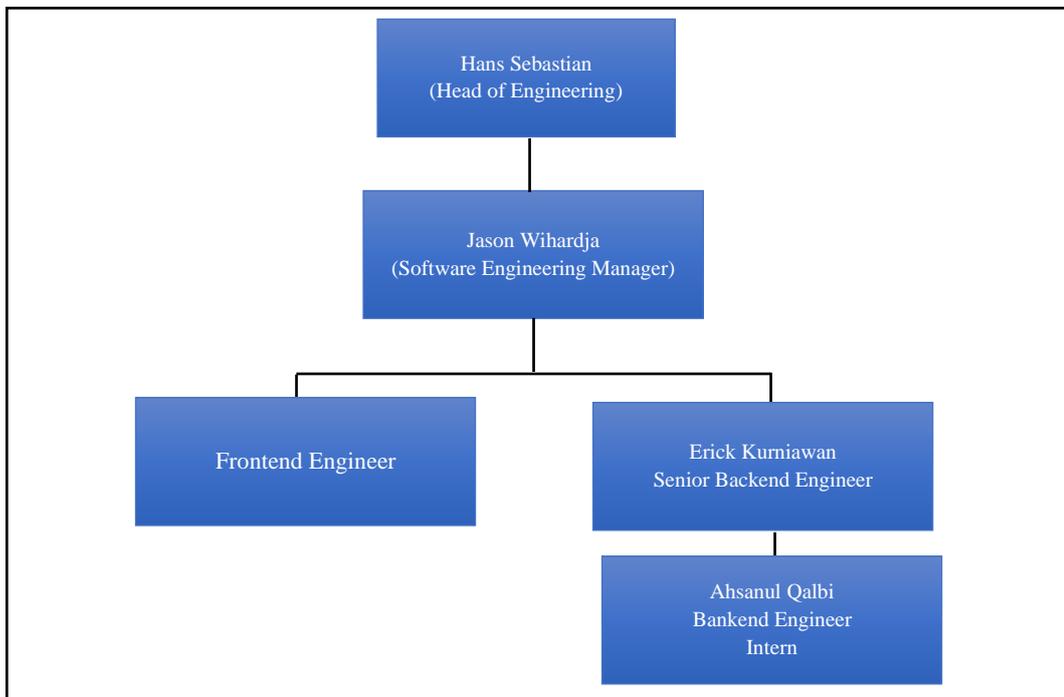
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang, penulis berperan sebagai *junior backend engineer intern* di divisi *Engineering* tim Account Management, dibimbing oleh Erick Kurniawan selaku *senior backend engineer*. Tugas diberikan oleh Bapak Jason Wihardja selaku *software engineering manager* melalui *sprint planning*. Susunan kedudukan dan koordinasi dalam pelaksanaan kerja magang digambarkan pada Gambar 3.1 sebagai berikut.



Gambar 3.1 Kedudukan dan Koordinasi Pelaksanaan Magang

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, tanggung jawab yang diberikan adalah sebagai berikut :

- Menjadi bagian dari tim pengembangan produk
- Mengerjakan tugas-tugas implementasi software seperti yang ditugaskan oleh Senior Software Engineer
- Membuat Backend API untuk Mobile Apps
- Menambahkan test unit pada fitur yang sudah ada
- Menambahkan Tes Integrasi pada integrasi partnership

3.3 Uraian Pelaksanaan Kerja Magang

3.3.1 Metode Pengembangan Sistem

Pengembangan sistem *backend engine* JULO menggunakan metode pengembangan Agile. Dengan menggunakan metode Agile, memungkinkan untuk melakukan perubahan pada spesifikasi pengembangan fitur sesuai dengan kebutuhan pelanggan. Tahap tahap dalam pengembangan fitur terdiri dari *planning* (analisis kebutuhan dan desain), pembangunan fitur, dan uji coba fitur yang dikembangkan.

Tahap *planning* diawali oleh divisi product bersama dengan CEO merencanakan dan analisis mengenai fitur apa saja yang harus dikembangkan sesuai dengan kebutuhan pelanggan. Kemudian *product manager*

menyampaikan spesifikasi fitur pada *developer* melalui *sprint planning* yang bertujuan untuk menganalisis *requirement* dari fitur yang akan dikembangkan.

Tahap selanjutnya adalah tahap pengembangan fitur berupa *coding* sesuai dengan *requirement*. Setelah pengembangan, dilakukan tahap selanjutnya yaitu tahap uji coba yang dilakukan oleh *developer* dan QA di dalam *branch* GitHub yang digunakan untuk *testing*, jika tahap uji coba berhasil, maka fitur baru dapat dimasukkan ke *branch* GitHub yang digunakan untuk proses *release* fitur sesuai dengan jadwal yang ditentukan oleh *scrum master*. Keempat tahap tersebut terus di iterasi setiap minggunya sehingga terus dapat mengembangkan fitur-fitur baru di dalam sistem *backend engine* sesuai dengan kebutuhan pelanggan.

3.3.2 Proses Pelaksanaan

Kerja magang dilaksanakan selama 3 (tiga) bulan atau 12 minggu. Pelaksanaan kerja magang terdapat rincian tugas seperti pada Tabel 3.1.

Tabel 3.1 Deskripsi Tugas yang Dilakukan

Minggu ke	Deskripsi
1	<ul style="list-style-type: none">• Instalasi <i>software</i> yang diperlukan.• Mempelajari <i>framework</i> Django, penggunaan <i>software</i>, dan python unit test.• Membuat unit test pada beberapa fungsi dari sistem JULO Messaging Service.
2	<ul style="list-style-type: none">• Mempelajari REST API Django.• Mengerjakan tugas pembuatan REST API Summary View dan Report View.

Tabel 3.1 Deskripsi Tugas yang Dilakukan (lanjutan)

3	<ul style="list-style-type: none"> • Melanjutkan pengerjaan tugas pembuatan REST API Summary View dan Report View. • Mempelajari konsep Docker Container.
4 - 6	<ul style="list-style-type: none"> • Mengerjakan tugas Dockerize Juloserver dengan membuat dockerfile, docker-compose.
7	<ul style="list-style-type: none"> • Menguji hasil pengembangan tugas Dockerize Juloserver dengan tim QA, dan mempresentasikan hasil pengembangan dockerize juloserver kepada tim Engineering JULO.
8	<ul style="list-style-type: none"> • Mengerjakan tugas <i>sprint</i> “Add Application with INITIATED Status in 170 to Slackbot Report”.
9	<ul style="list-style-type: none"> • Mengerjakan tugas <i>sprint</i> “ T-3 remove agent intervention”
10	<ul style="list-style-type: none"> • Mengerjakan tugas <i>sprint</i> “ PEDE collection reminder”
11 - 13	<ul style="list-style-type: none"> • Mengerjakan tugas <i>sprint</i> “Bukalapak invoice ID in CRM” • Mengerjakan tugas <i>sprint</i> “Repayment Notification through Google Calendar”.
14	<ul style="list-style-type: none"> • Mengerjakan tugas <i>sprint</i> “Add readiness dependencies in julo server docker compose”.

Pada minggu pertama dilakukan instalasi *software* yang diperlukan untuk pengembangan program, mempelajari *framework* yang digunakan oleh PT JULO yaitu Django, mempelajari *unit test* python, dan mengerjakan tugas berupa menambahkan *unit test* pada fungsi fungsi pada sistem JULO Messaging Service.

Pada minggu kedua dilakukan pembelajaran pada Django REST API, serta dilakukan pengerjaan tugas “Add Summary and Report View API on Follow The Money” berupa pembuatan REST API untuk digunakan oleh divisi *frontend* berupa data peminjaman pelanggan yang diambil dari *database*.

Pada minggu ketiga dilakukan lanjutan pengerjaan tugas “Add Summary and Report View API on Follow The Money”. Pada minggu ini pun dilakukan pembelajaran terhadap konsep Docker Container.

Pada minggu keempat sampai minggu ketujuh dilakukan pengerjaan tugas untuk membuat Docker Image untuk sistem perusahaan yang akan dijalankan pada Docker Container yang merupakan wadah untuk menjalankan Docker Image atau *blueprint* dari sistem perusahaan, dan dilakukan presentasi kepada tim Engineering mengenai hasil penggunaan Docker pada sistem perusahaan dan cara menjalankan *juloserver* dengan menggunakan Docker Container.

Pada minggu kedelapan dilakukan pengerjaan tugas yang didapat dari *sprint planning* yaitu “Add Application with INITIATED Status in 170 to Slackbot Report” penambahan pemasukan pada Slackbot Report berupa aplikasi pelanggan yang telah menandatangani perjanjian hukum namun mengalami penundaan pencairan dana, dengan membuat *cron task*.

Pada minggu kesembilan dilakukan pengerjaan tugas yang didapat dari *sprint planning* yaitu “T-3 remove agent intervention”. Tugas ini berupa penggantian notifikasi pembayaran pada tiga hari sebelum jatuh tempo yang sebelumnya jika pelanggan gagal menerima pemanggilan Robocall sebanyak 3 kali maka dilakukan pemanggilan telepon oleh divisi Agent kemudian jika pemanggilan gagal maka pelanggan akan mendapatkan pesan melalui Whatsapp, melalui tugas ini, prosedur tersebut diubah menjadi jika pelanggan gagal menerima pemanggilan oleh Robocall sebanyak 3 kali, maka akan langsung mendapat pesan melalui

Whatsapp, dan Whatsapp tersebut dikirimkan sesuai dengan zona wilayah tempat tinggal pelanggan.

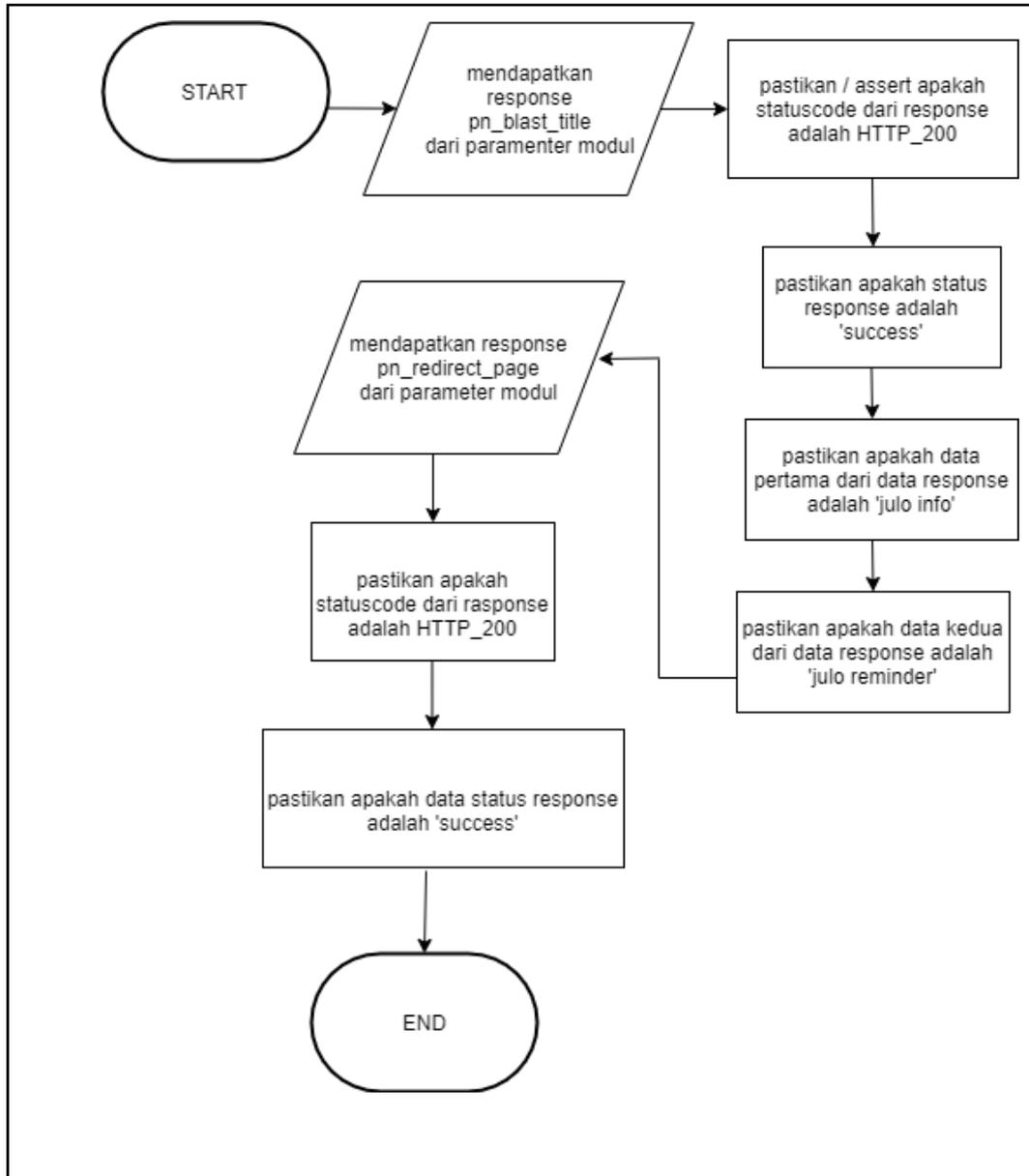
Pada minggu kesepuluh dilakukan pengerjaan tugas yang didapat dari *sprint planning* yaitu “PEDE collection reminder” yaitu pengembangan dari tugas sebelumnya di minggu ke sembilan berupa notifikasi tambahan yaitu peminjaman jenis “Ponsel Duit” kepada pelanggan beserta *template* pesan Whatsapp nya.

Pada minggu kesebelas sampai minggu ketiga belas dilakukan pengerjaan pada tugas “Bukalapak invoice ID in CRM” berupa penambahan kolom *invoice_ID* pelanggan yang menggunakan aplikasi partner Bukalapak pada tabel di situs CRM, dan dilakukan pengerjaan tugas “Repayment Notification through Google Calendar” berupa penambahan *attachment file* dengan format .ics pada fungsi yang bertugas untuk mengirimkan email notifikasi pada pelanggan ketika dana pinjaman telah sukses dicairkan. File ics atau iCalendar tersebut berisi keterangan pembayaran dan jadwal jatuh tempo pembayaran pelanggan yang diambil dari *database*.

Pada minggu terakhir yakni minggu keempat belas, dilakukan pengerjaan pada tugas “Add readiness dependencies in julio server docker compose” berupa penambahan fitur pada tugas di minggu ke 4 sampai 7 yang berkaitan dengan *dockerfile*. Fitur tersebut merupakan fitur untuk menambahkan *readiness dependencies* pada service docker container. Sehingga suatu container tidak akan dijalankan lebih dahulu dari container harus dijalankan terlebih dahulu.

3.3.3 Perancangan Sistem

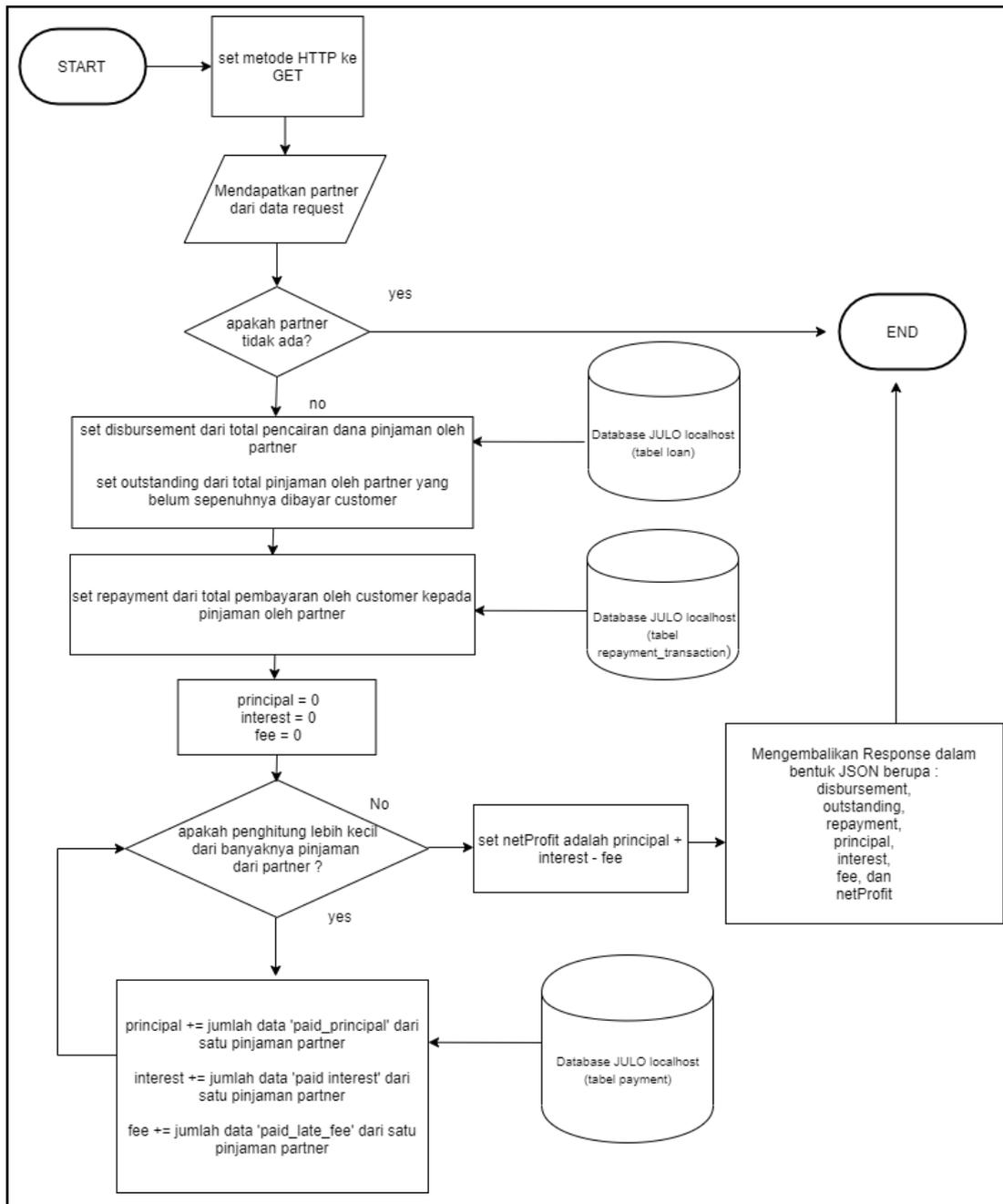
A. Flowchart



Gambar 3.2 Flowchart tugas unit test pada JULO messaging service

Gambar 3.2 merupakan *flowchart* dari pengerjaan tugas untuk menambahkan python unit test pada fungsi `pn_blast_title` dan `pn_redirect_page` pada aplikasi Julo Messaging Service. Pertama dilakukan pengambilan data response `pn_blast_title` yang merupakan parameter dari fungsi unit test. Setelah itu dilakukan *checking* terhadap isi dari statuscode dari response, serta dilakukan pengecekan pada isi data response pertama dan kedua apakah berisi 'julo info' dan 'julo reminder'.

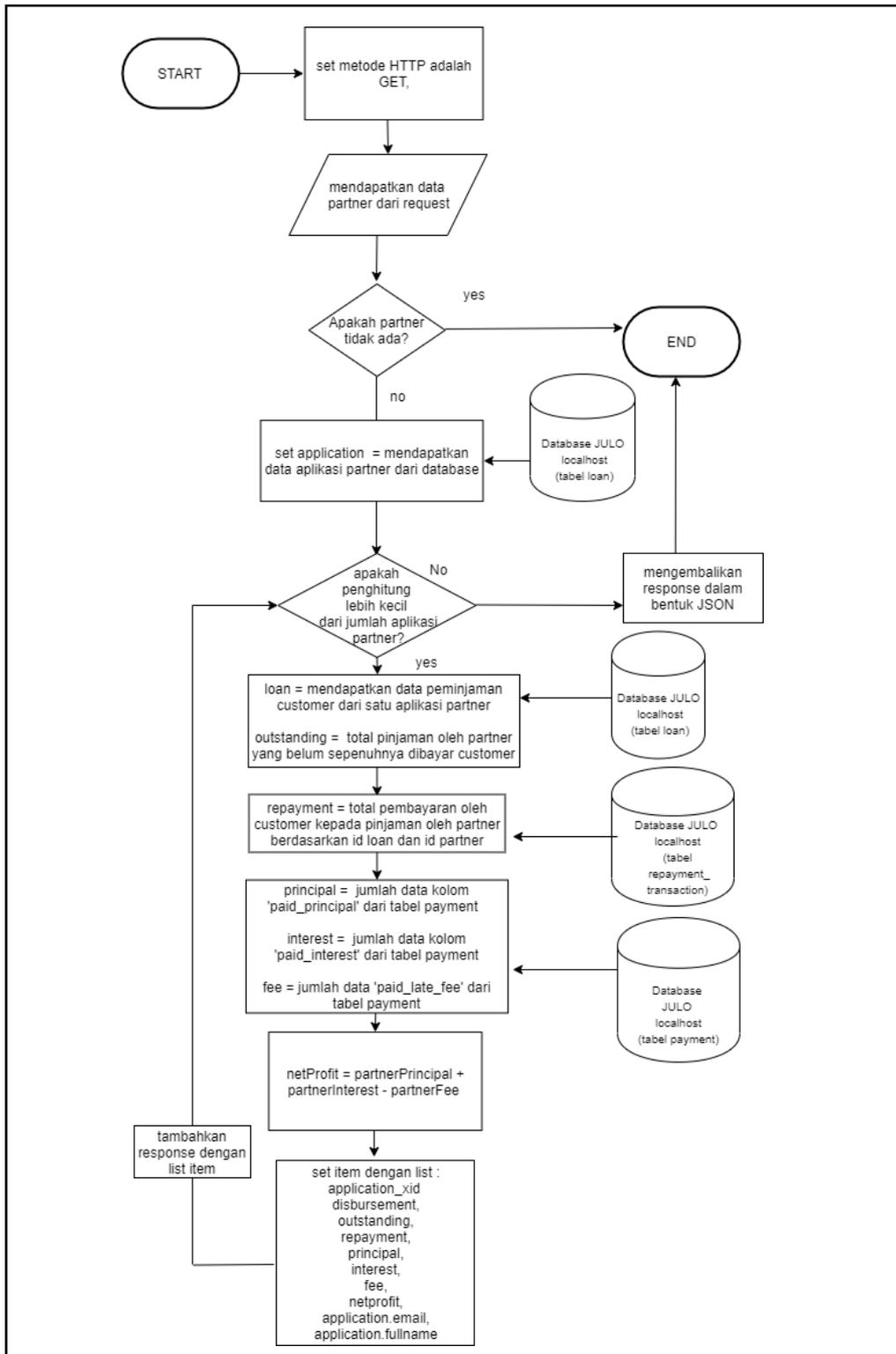
Selanjutnya dilakukan pengambilan data response `pn_redirect_page` yang merupakan parameter dari fungsi unit test, dan dilakukan pengecekan pada isi dari statuscode response apakah berisi `HTTP_200_OK`. Jika semua variable yang di cek isinya sesuai dengan yang diharapkan, maka fungsi tersebut lulus python unit test.



Gambar 3.3 Flowchart Summary View REST API

Gambar 3.3 merupakan *flowchart* dari pengerjaan tugas “API Summary View”. Pertama dilakukan penetapan HTTP method dengan GET. Selanjutnya dilakukan pengambilan data partner dari data request melalui pengambilan data pada *user login*. Lalu dilakukan pengecekan apakah terdapat data partner dari request tersebut, jika terdapat data partner, maka proses dilanjutkan pada pengambilan data peminjaman partner berupa *disbursement*, *outstanding*, dan *repayment* dari *database*.

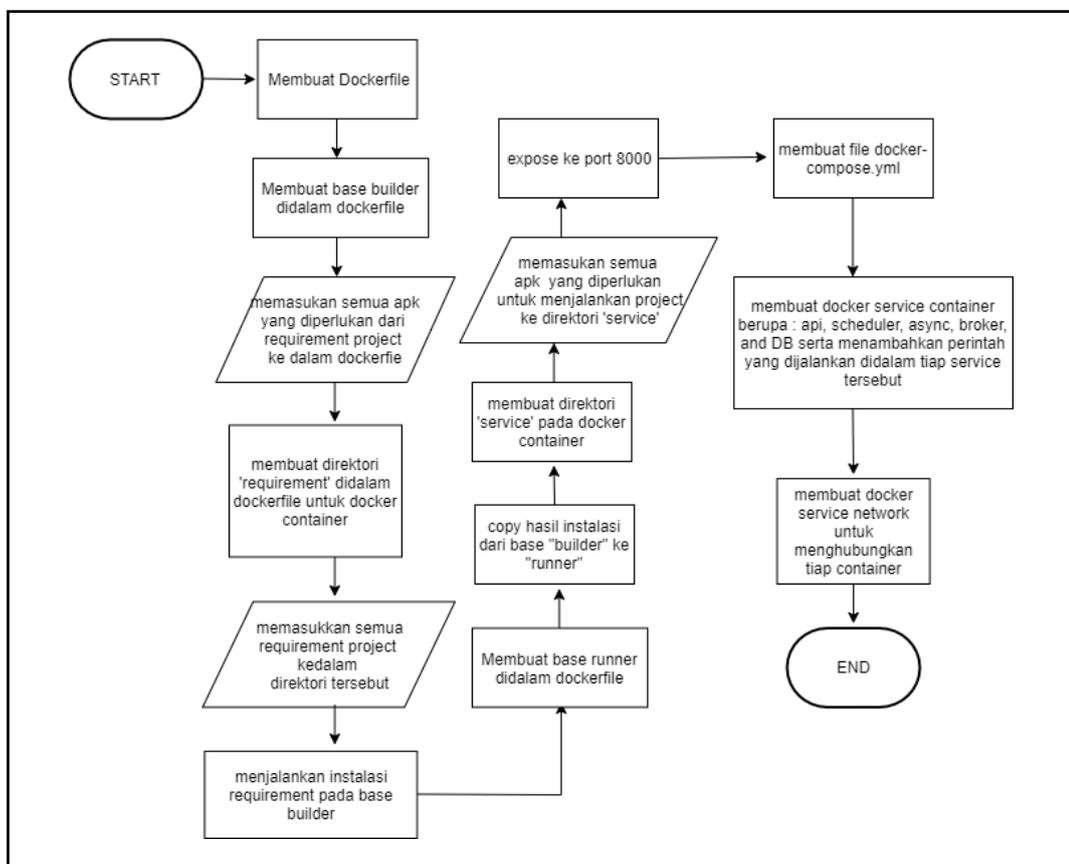
Selanjutnya dilakukan *looping* untuk menghitung total data dari data pinjaman pengguna pada suatu partner seperti *principal*, *interest*, dan *fee*. Data-data tersebut diambil dari tabel payment berdasarkan data loan dari *partner* tersebut. Data-data tersebut akan di total sesuai dengan banyaknya data pinjaman dari *partner*. Setelah proses *looping* selesai dilakukan pengembalian beberapa data pinjaman pelanggan dalam bentuk response JSON. Response tersebut nantinya akan digunakan oleh divisi *frontend* untuk menampilkan data-data tersebut pada halaman beranda untuk penyedia dana,



Gambar 3.4 Flowchart Report View REST API

Gambar 3.4 merupakan *flowchart* dari pengerjaan tugas “API Report View”. Pertama dilakukan penetapan HTTP method dengan GET. Selanjutnya dilakukan pengambilan data partner dari data request. Lalu dilakukan pengambilan data dari *database* berupa data aplikasi pelanggan yang menggunakan pinjaman dari partner yang diambil dari data request.

Selanjutnya dilakukan *looping* yang bertujuan untuk mengambil data-data dari satu pinjaman pelanggan dari satu aplikasi partner tersebut. Selanjutnya data tersebut ditambahkan pada list response yang akan dikembalikan dalam bentuk JSON melalui URL yang telah ditentukan.

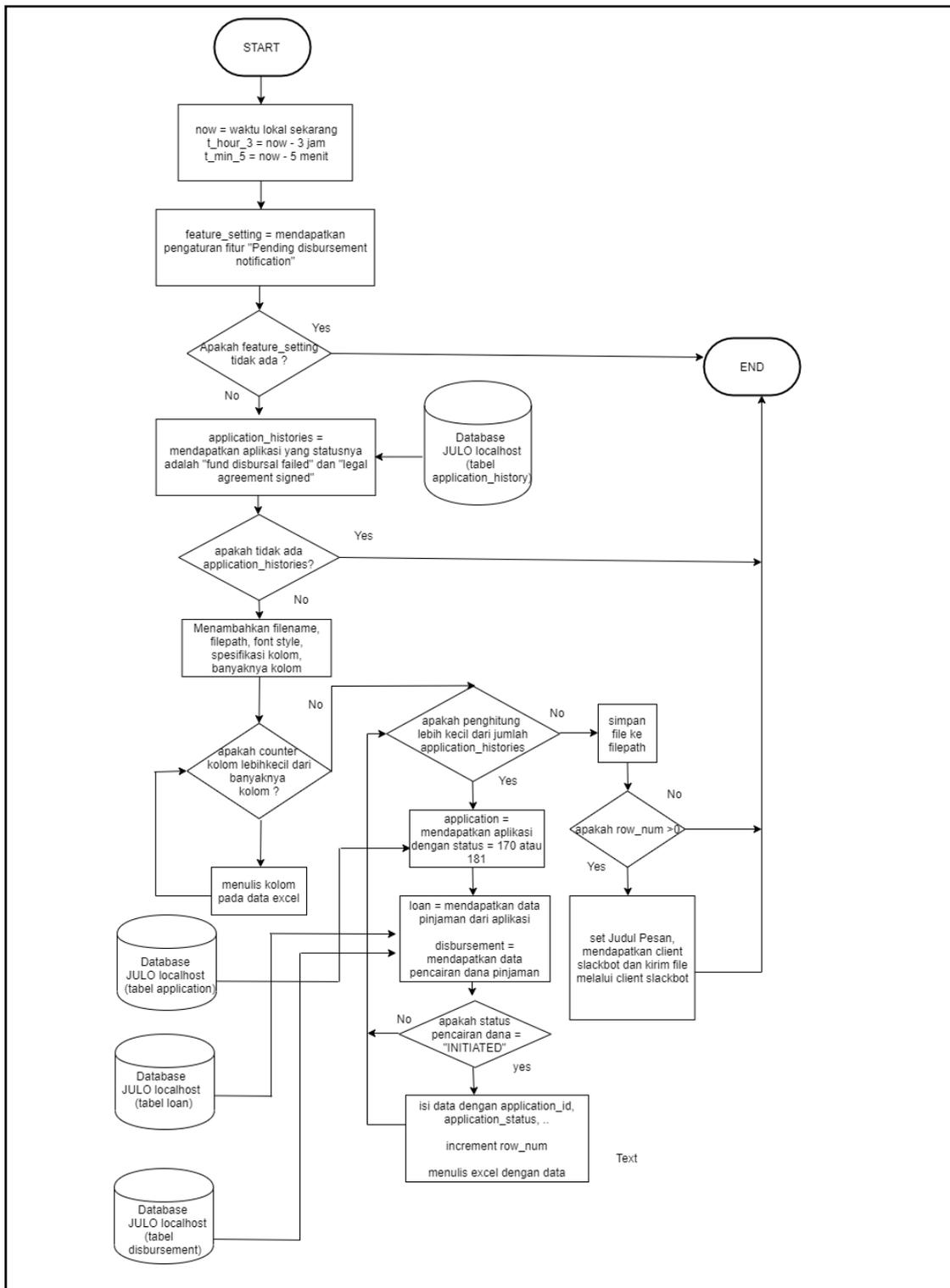


Gambar 3.5 Flowchart Tugas Dockerize Juloserver

Gambar 3.5 merupakan *flowchart* dari pengerjaan tugas “Dockerize Juloserver”. Pertama dilakukan pembuatan Dockerfile dan pembuatan base *builder* yang berguna untuk memasukan semua apk yang diperlukan pada requirement project, pada base tersebut juga dilakukan duplikasi semua *project requirement* pada direktori *requirement* pada docker container. Selanjutnya menjalankan instalasi pada base builder.

Selanjutnya dibuat base *runner* pada dockerfile serta mengkopi instalasi yang dibuat oleh base *builder* pada base *runner*. Membuat direktori baru bernama *service*, lalu memasukan semua apk yang diperlukan untuk menjalankan sistem pada direktori *service*. Selanjutnya dilakukan ekspose port ke port 8000.

Setelah pembuatan dockerfile, selanjutnya dilakukan pembuatan file docker-compose. Pada docker-compose, dilakukan pembuatan docker container service berupa container API, database, scheduler, async, dan broker. Lalu ditambahkan *network service* pada docker-compose untuk menghubungkan masing masing service container.

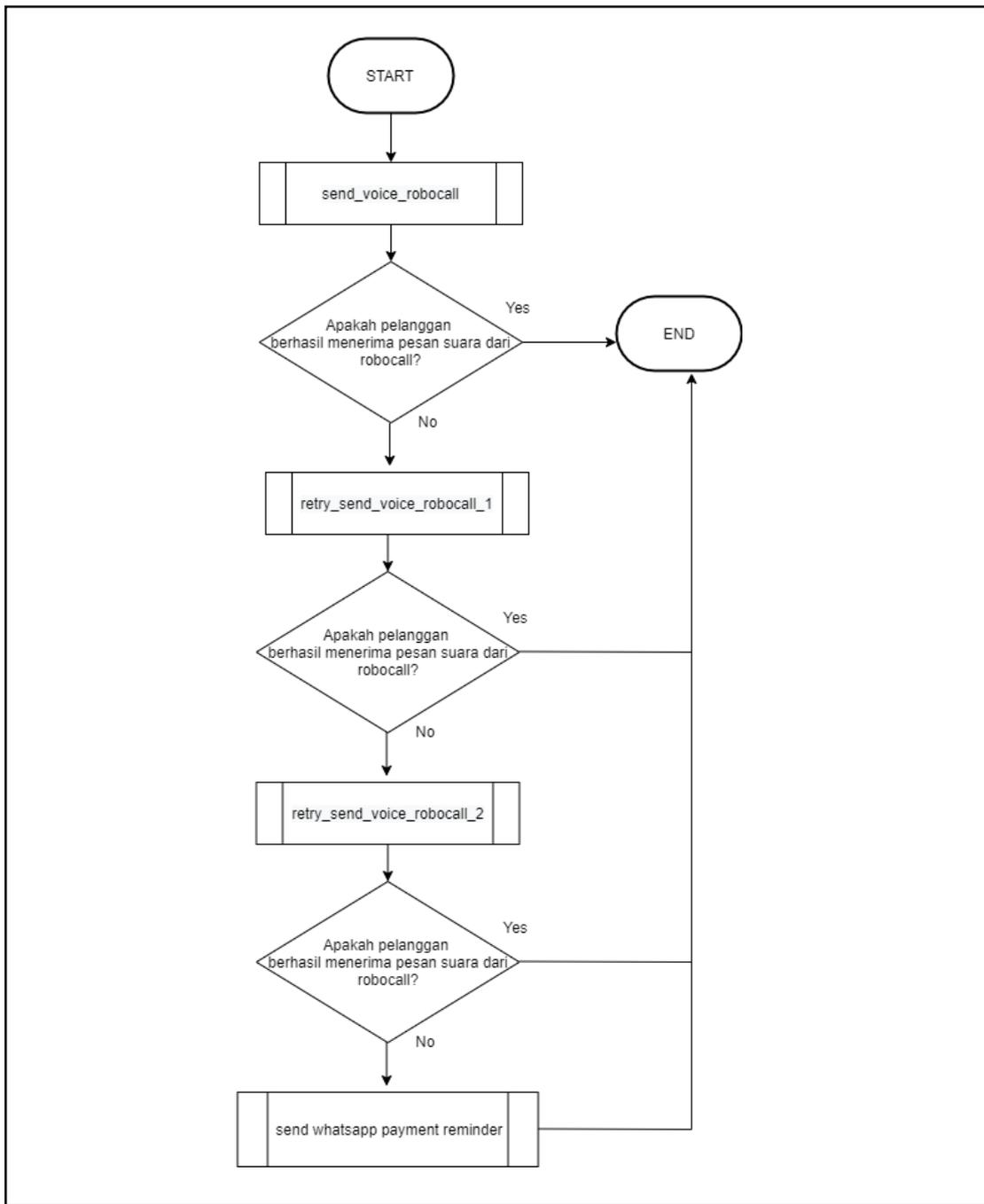


Gambar 3.6 Flowchart tugas Add Application with INITIATED Status in 170 to Slackbot Report

Gambar 3.6 merupakan *flowchart* dari pengerjaan tugas “Add Application with INITIATED Status in 170 to Slackbot Report”. Pertama dilakukan pengambilan waktu lokal dan pengambilan pengaturan fitur berupa format untuk dokumen Pending Disbursement Notification.

Selanjutnya dilakukan pengambilan data riwayat aplikasi dari database dengan status Fund Disbursal Failed atau Legal Agreement Signed. Jika terdapat riwayat aplikasi maka proses dilanjutkan dengan mespesifikasi nama file, *filepath*, spesifikasi kolom Excel, dan banyaknya kolom.

Selanjutnya dilakukan *looping* untuk menulis kolom *header* pada Excel. Lalu dilakukan *looping* untuk mendapatkan data pinjaman pelanggan dengan status 170 dan status 181. Lalu didapatkan data pencairan dana terhadap pinjaman tersebut. Jika status pencairan dana tersebut adalah INITIATED, maka data tersebut akan ditulis pada Excel. Selanjutnya akan di tetapkan judul pesan dan deskripsi pesan serta mengirimkan file Excel tersebut ke aplikasi Slack melalui client slackbot.

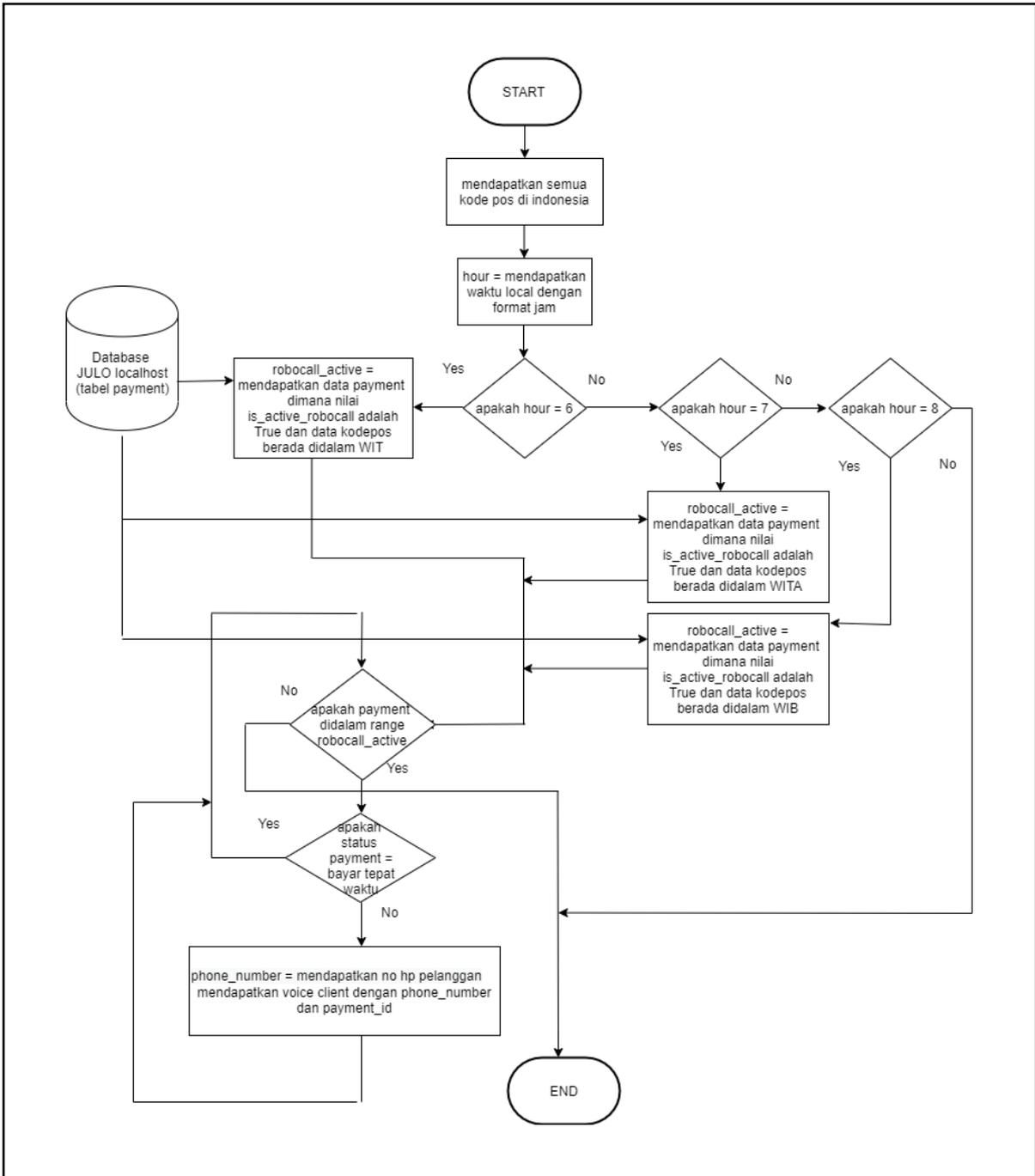


Gambar 3.7 Flowchart main module tugas t-3 remove agent intervention

Gambar 3.7 merupakan *flowchart* modul utama dari pengerjaan tugas “remove agent intervention”. Pertama main modul akan memanggil submodul `send_voice_robotcall`. Submodul tersebut berfungsi untuk melakukan pengiriman pesan suara pada pelanggan melalui layanan robocall JULO. Pesan tersebut bertujuan untuk memberi tahu pelanggan bahwa peminjaman mereka akan jatuh tempo pembayaran pada tiga hari kedepan. Submodul tersebut dijalankan secara otomatis melalui *cron job* pada jam delapan pagi sesuai dengan zona waktu pelanggan.

Selanjutnya jika pelanggan tidak berhasil menerima pesan suara robocall, maka akan dilakukan pemanggilan submodul `retry_send_voice_robotcall_1`, dan `retry_send_voice_robotcall_2`. Kedua submodul tersebut berfungsi sama seperti `send_voice_robotcall`, namun submodul tersebut dijalankan pada jam sepuluh pagi dan jam duabelas siang sesuai dengan zona waktu pelanggan. Jika ketiga modul robocall tersebut telah dijalankan, namun pelanggan tetap tidak berhasil menerima pesan suara dari robocall, maka akan dilakukan pemanggilan submodul `send_whatapp_payment_reminder`.

Submodul `send_whatapp_payment_reminder` berfungsi untuk melakukan pemberian pesan kepada pelanggan melalui aplikasi whatsapp. Sama seperti pengiriman pesan robocall, pengiriman pesan whatsapp bertujuan untuk memberi tahu pelanggan bahwa peminjaman mereka akan jatuh tempo pembayaran pada tiga hari kedepan, serta memberikan informasi mengenai perincian pinjaman pelanggan serta informasi mengenai cara membayar peminjaman.

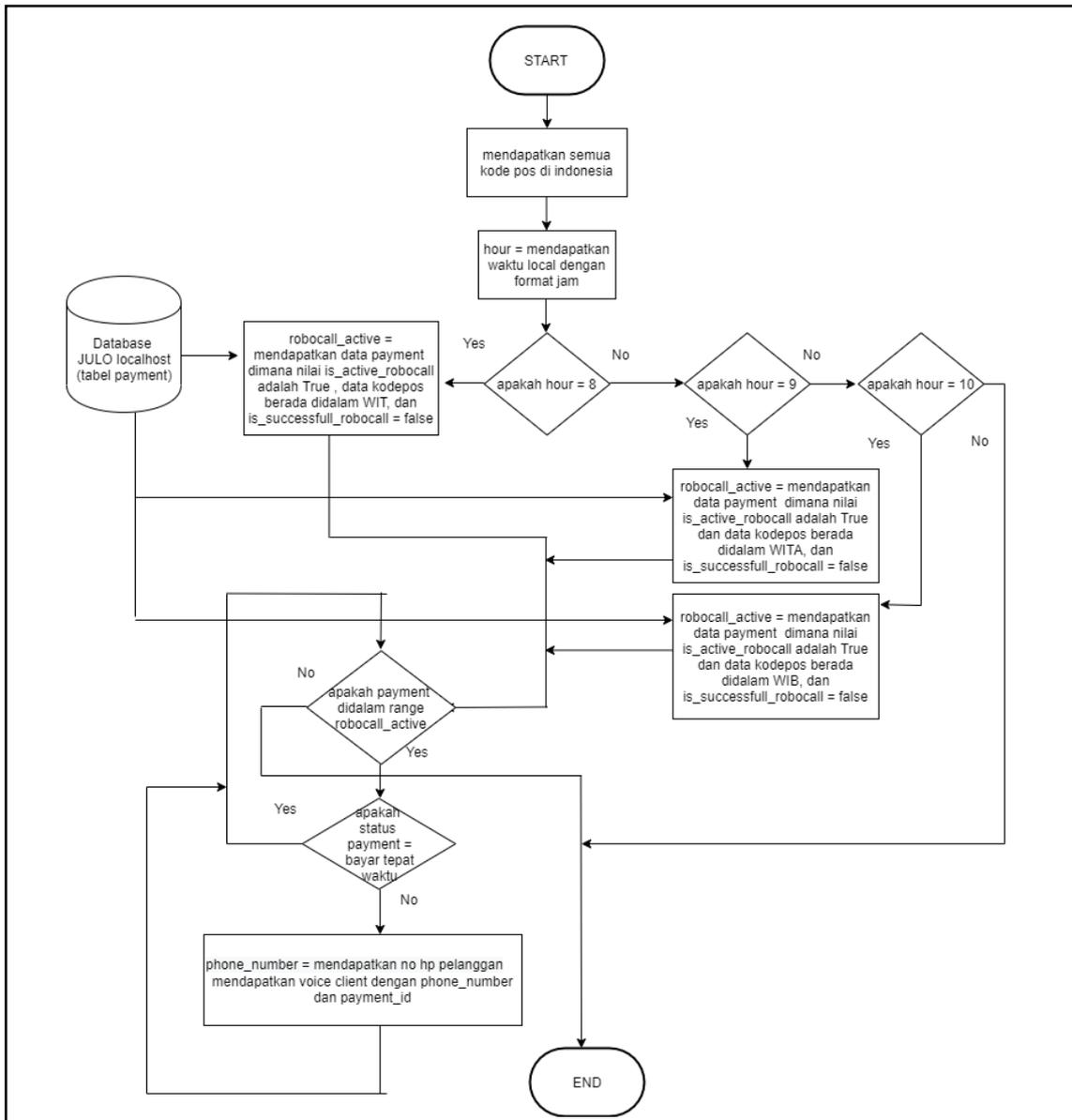


Gambar 3.8 Flowchart send_voice_robocall

Gambar 3.8 merupakan *flowchart* dari submodule `send_voice_robo`. Modul ini dijalankan oleh *cron job* setiap jam enam, tujuh, dan delapan pagi sesuai dengan zona waktu server JULO. Pertama dilakukan pengambilan data kode pos seluruh Indonesia pada file *project* yang berfungsi untuk menyimpan variabel konstan. Selanjutnya dilakukan pengambilan waktu lokal pada server JULO dengan format jam.

Lalu proses dilanjutkan dengan pengecekan waktu jam lokal. Jika waktu jam lokal server JULO adalah jam 6 pagi, maka akan dilakukan pengambilan data pada table *payment* berupa data peminjaman pelanggan dengan nilai kolom `is_robo` adalah `True` dan pelanggan dengan zona waktu WIT. Namun jika waktu jam lokal server JULO adalah jam 7 pagi, maka akan dilakukan pengambilan data dengan nilai kolom `is_robo` adalah `True` dan pelanggan dengan zona waktu WITA. Kondisi terakhir jika waktu jam lokal server JULO adalah jam 8 pagi, maka akan dilakukan pengambilan data dengan nilai kolom `is_robo` adalah `True` dan pelanggan dengan zona waktu WIB.

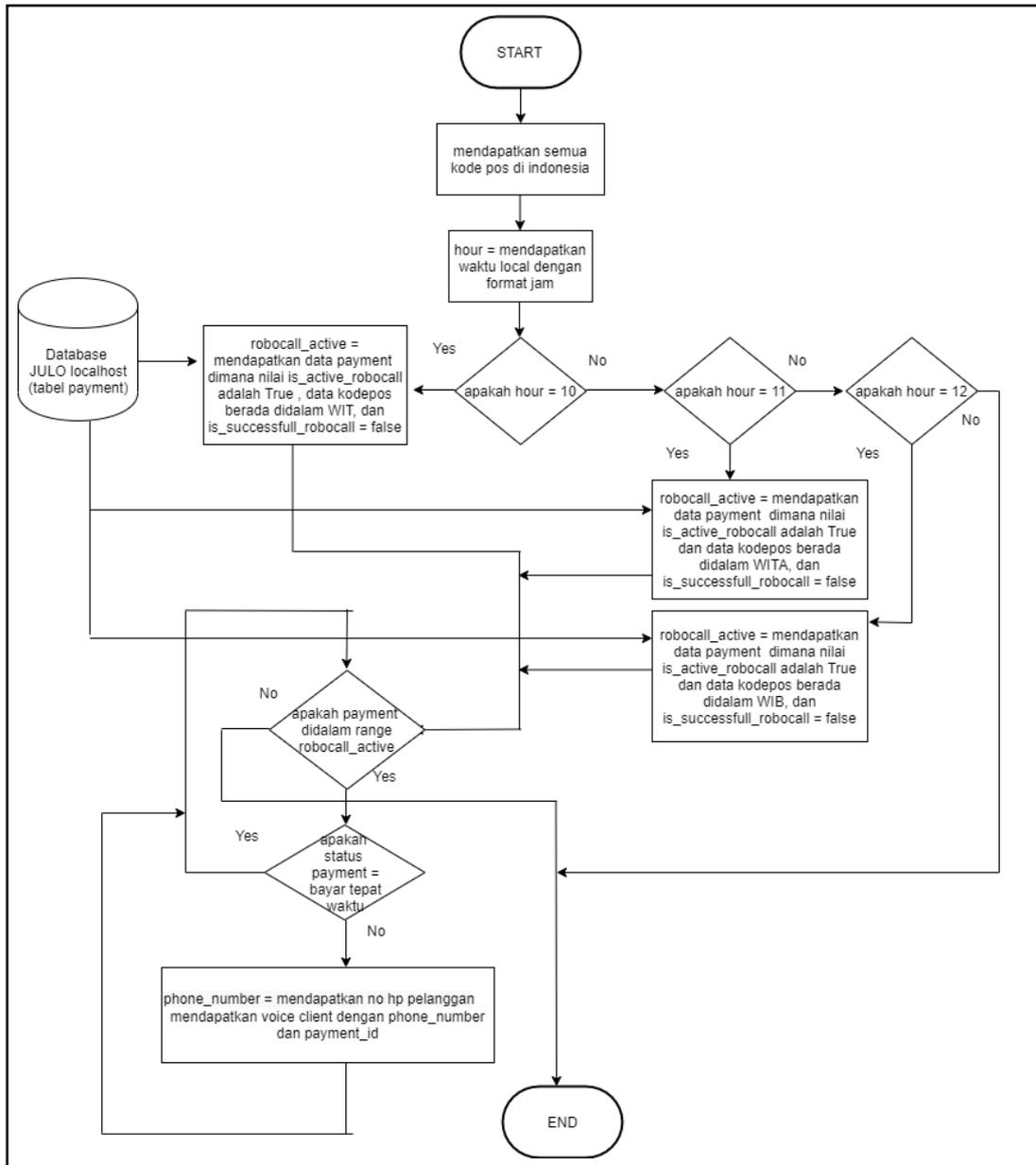
Selanjutnya dilakukan *looping* dengan menelusuri list seluruh data pinjaman yang didapat pada proses sebelumnya. Jika data pinjaman memiliki status dibayar tepat waktu, maka *looping* akan dilanjutkan ke data *payment* selanjutnya, namun jika tidak, maka akan dilakukan proses pengiriman pesan suara dengan menggunakan *robo*.



Gambar 3.9 Flowchart retry_send_voice_robocall_1

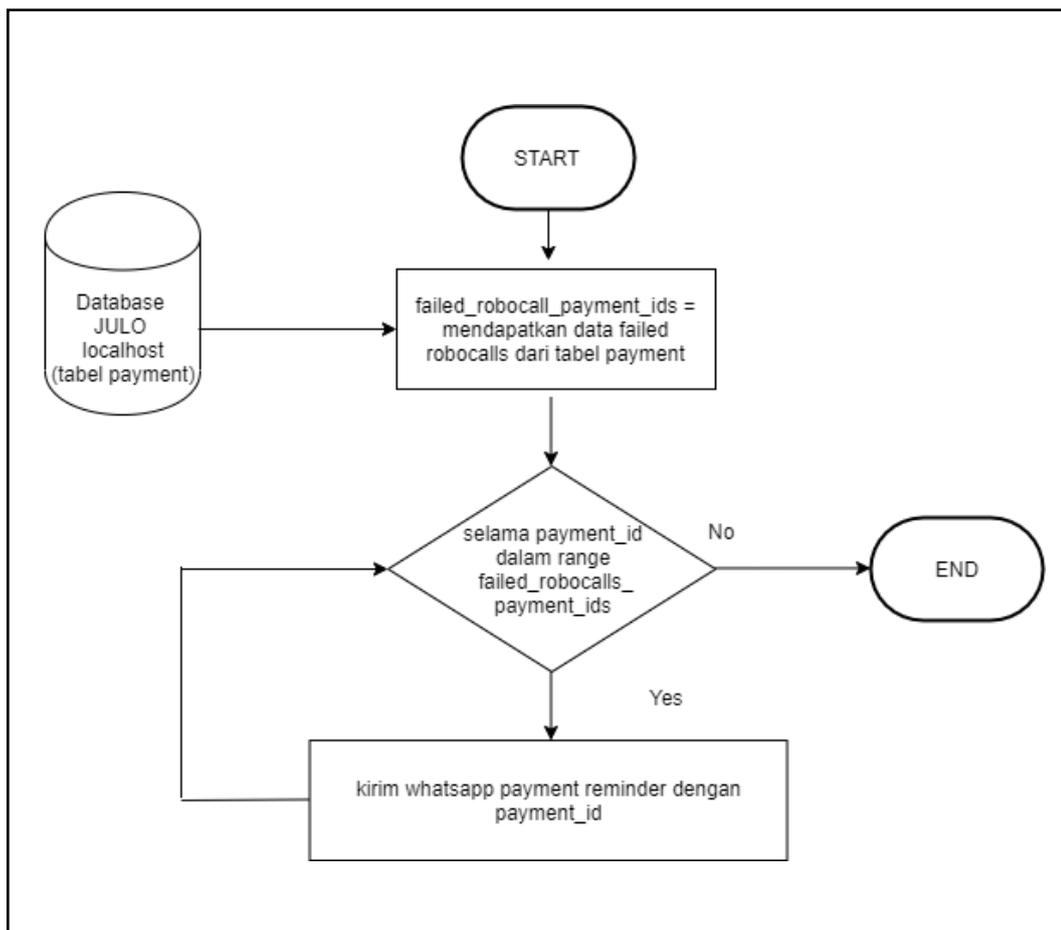
Gambar 3.9 merupakan *flowchart* dari submodul `retry_send_voice_robocall_1`. Submodul tersebut memiliki fungsi yang serupa dengan `send_voice_robocall`, namun dengan kondisi pengiriman yang berbeda yaitu dengan pengecekan waktu lokal pada server JULO apakah memiliki value jam delapan, sembilan, dan jam sepuluh pagi, dikarenakan modul tersebut bertujuan

untuk mengirimkan pesan suara robocall pada pukul sepuluh pagi sesuai dengan zona waktu pelanggan. Serta memiliki perbedaan pada pengambilan data dari tabel payment yaitu dengan kondisi tambahan berupa is_successfull_robocall adalah false.



Gambar 3.10 Flowchart retry_send_voice_robocall_2

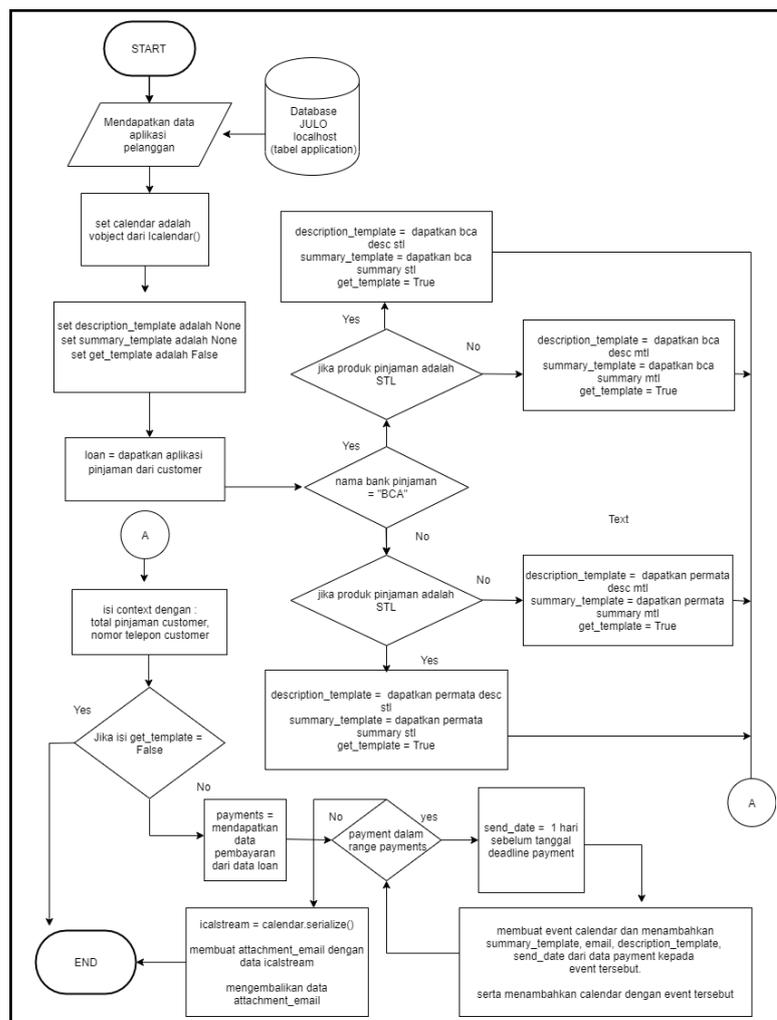
Gambar 3.10 merupakan *flowchart* dari submodul `retry_send_voice_robocall_2`. Submodul tersebut memiliki fungsi yang serupa dengan `retry_send_voice_robocall_1`, namun dengan kondisi pengiriman yang berbeda yaitu dengan pengecekan waktu lokal pada server JULO apakah memiliki value jam sepuluh, sebelas, dan jam duabelas, dikarenakan modul tersebut bertujuan untuk mengirimkan pesan suara robocall pada pukul sepuluh pagi sesuai dengan zona waktu pelanggan.



Gambar 3.11 Flowchart `send_whatsapp_payment_reminder`

Gambar 3.11 merupakan *flowchart* submodul

send_whatapp_payment_reminder. Pertama dilakukan penarikan data dari tabel payment berupa *row* yang memiliki kondisi is_successfull_robocall dengan nilai false. Lalu dilakukan *looping* dengan menelusuri list seluruh data payment id yang didapat dari proses sebelumnya. Di dalam proses *looping* dilakukan pengiriman whatsapp sesuai dengan payment_id yang digunakan untuk mengambil data payment untuk dilampirkan pada pesan whatsapp.



Gambar 3.12 Flowchart pengerjaan tugas Repayment Through Google Calendar

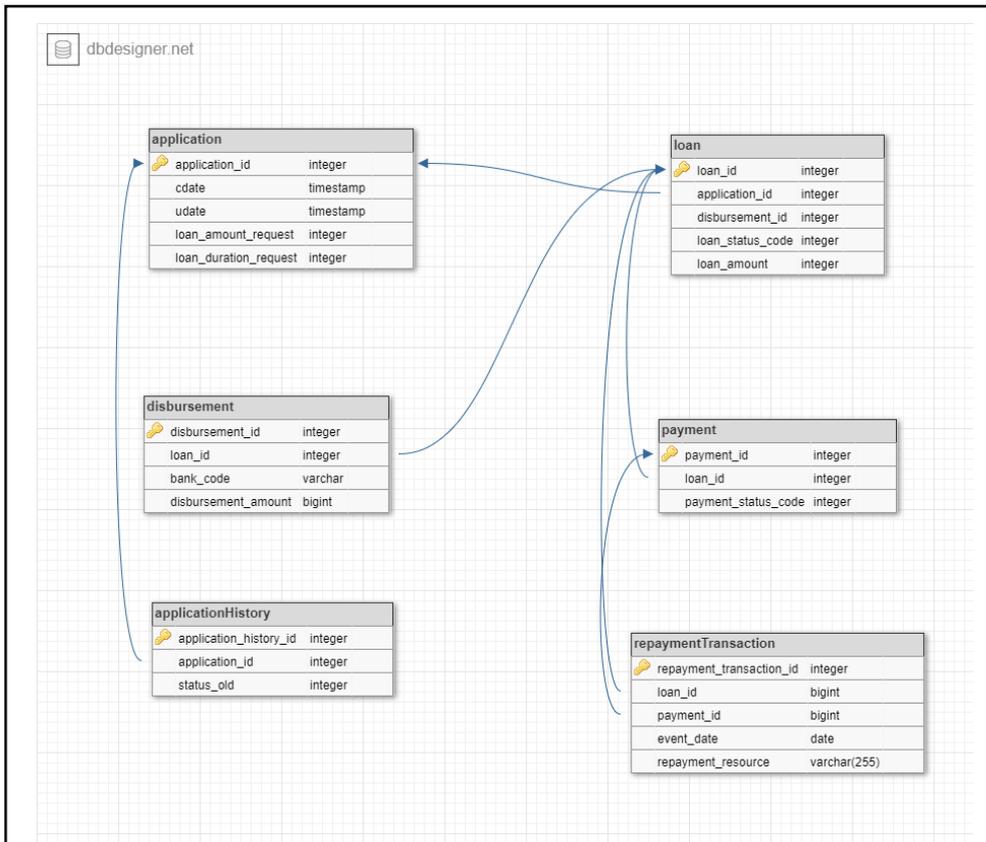
Gambar 3.12 merupakan *flowchart* dari pengerjaan tugas “Repayment Through Google Calendar”. Dimulai dengan pengambilan data aplikasi pelanggan yang ingin dikirim email dengan *attachment* notifikasi google calendar. Lalu dilanjutkan dengan pembuatan objek calendar sebagai vobject icalendar dan pengambilan data pinjaman pelanggan dari data aplikasi pelanggan.

Setelah proses dilanjutkan dengan menetapkan *template* deskripsi dan ringkasan pesan yang akan dikirim pada notifikasi google calendar sesuai dengan nama bank dan produk pinjaman yang digunakan oleh pelanggan. Selanjutnya dilakukan pengisian variabel konteks yang digunakan untuk mengisi data pada *template* deskripsi notifikasi.

Selanjutnya jika nilai *get_template* adalah true, maka proses berlanjut pada pengambilan data pada table payment dengan menggunakan relasi dengan tabel loan berupa data keterangan pembayaran pelanggan. Lalu dilakukan *looping* dengan menelusuri list seluruh data payment yang didapat dari proses sebelumnya. Didalam *looping* tersebut dilakukan pembuatan *event* baru pada object calendar dengan menyertakan data keterangan notifikasi, alamat email, serta tanggal pengiriman notifikasi.

Tahap selanjutnya adalah membuat objek baru berupa serialisasi dari objek calendar, serta membuat *dictionary* yang digunakan sebagai *attachment* email dengan menyertakan file calendar serta mengembalikan data *attachment* tersebut.

B. Skema Database



Gambar 3.13 Skema Database Pengerjaan Tugas Magang

Pada Gambar 3.13, terdapat 5 tabel pada *database* JULO yang digunakan dalam pengerjaan dan pengembangan tugas yang diberikan selama kerja magang. Tabel-tabel yang digunakan antara lain adalah tabel application, tabel loan, tabel disbursement, tabel payment, tabel repaymentTransaction dan tabel applicationHistory. Kelima tabel tersebut merupakan tabel yang berfungsi untuk menyimpan data-data pelanggan. Data-data yang disimpan antara lain data registrasi pelanggan, data pencairan dana, riwayat peminjaman, data pembayaran dan data peminjaman pelanggan.

C. Struktur Tabel

a. Tabel application

Gambaran struktur tabel application berupa nama kolom dan tipe data terdapat pada Tabel 3.2. Tabel application berfungsi untuk menyimpan data *user*.

Tabel 3.2 Struktur Tabel application

Nama Kolom	Tipe Data	Ukuran	Keterangan
cdate	timestamp	35	Waktu saat data dibuat
update	timestamp	35	Waktu saat data <i>update</i>
application_id	int8	19	ID aplikasi <i>user</i>
loan_amount_request	int8	19	Jumlah pengajuan pinjaman
loan_duration_request	int4	10	Durasi pinjaman
loan_purpose	varchar	100	Tujuan peminjaman
marketing_source	varchar	100	Sumber <i>user</i> mengetahui JULO
referral_code	varchar	20	Kode <i>referral</i>
is_own_phone	bool	1	Apakah <i>user</i> memiliki ponsel
fullname	varchar	100	Nama lengkap <i>user</i>
dob	date	13	Tanggal lahir <i>user</i>
gender	varchar	10	Jenis kelamin <i>user</i>
ktp	varchar	16	NIK <i>user</i>
address_street_num	varchar	100	Alamat Nomor jalan <i>user</i>
address_provinsi	varchar	100	Alamat Provinsi
address_kabupaten	varchar	100	Alamat Kabupaten atau kota
address_kecamatan	varchar	100	Alamat Kecamatan
address_kelurahan	varchar	100	Alamat Kelurahan
address_kodepos	varchar	5	Alamat Kode pos
occupied_since	date	13	Tanggal <i>user</i> menempati alamat yang dimaksud
home_status	varchar	50	Status kepemilikan rumah

Tabel 3.2 Struktur Tabel application (lanjutan)

Nama Kolom	Tipe Data	Ukuran	Keterangan
landlord_mobile_phone	varchar	50	Nomor telepon rumah
mobile_phone_1	varchar	50	Nomor ponsel utama
has_whatsapp_1	bool	1	Status whatsapp ponsel utama
mobile_phone_2	varchar	50	Nomor ponsel kedua
has_whatsapp_2	bool	1	Status whatsapp ponsel kedua
email	varchar	254	Email <i>user</i>
bbm_pin	varchar	50	Pin BBM
marital_status	varchar	50	Status pernikahan
dependent	int4	10	Tanggungans
spouse_name	varchar	100	Nama orang tua
spouse_dob	date	13	Tanggal lahir orang tua
spouse_mobile_phone	varchar	50	Nomor HP orang tua
spouse_has_whatsapp	bool	1	Kepemilikan whatsapp
kin_name	varchar	100	Nama saudara kandung
kin_dob	date	13	Tanggal lahir saudara kandung
kin_gender	varchar	10	Jenis kelamin saudara kandung
kin_mobile_phone	varchar	50	Nomor HP saudara kandung
kin_relationship	varchar	50	Jenis hubungan saudara kandung dengan <i>user</i>
job_type	varchar	50	Jenis pekerjaan yang dimiliki
job_description	varchar	100	Pekerjaan yang dimiliki <i>user</i>
company_name	varchar	100	Nama perusahaan di mana <i>user</i> bekerja
company_phone_number	varchar	50	Nomor telepon perusahaan
job_start	date	13	Tanggal <i>user</i> mulai bekerja
monthly_income	int8	19	Total pendapatan bulanan
income_1	int8	19	Detail pendapatan pertama

Tabel 3.2 Struktur Tabel application (lanjutan)

Nama Kolom	Tipe Data	Ukuran	Keterangan
income_2	int8	19	Detail pendapatan kedua
income_3	int8	19	Detail pendapatan ketiga
last_education	varchar	50	Pendidikan terakhir <i>user</i>
graduation_year	int4	10	Tahun lulus <i>user</i>
gpa	float8	17	IPK <i>user</i> (jika S1 ke atas)
has_other_income	bool	1	Status pendapatan sampingan
other_income_amount	int8	19	Jumlah pendapatan sampingan
other_income_source	varchar	250	Sumber pendapatan lain <i>user</i>
monthly_housing_cost	int8	19	Biaya sewa rumah
monthly_expenses	int8	19	Pengeluaran rumah tangga
total_current_debt	int8	19	Cicilan yang masih ditanggung
vehicle_type_1	varchar	50	Jenis kendaraan <i>user</i>
vehicle_ownership_1	varchar	50	Status kepemilikan kendaraan
bank_name	varchar	250	Nama bank yang digunakan
bank_branch	varchar	100	Cabang bank
bank_account_number	varchar	50	Nomor rekening <i>user</i>
is_term_accepted	bool	1	Status persetujuan form <i>user</i>
is_term_verification_agreed	bool	1	Status verifikasi data <i>user</i>
is_document_submitted	bool	1	Status pengumpulan dokumen
is_sphp_signed	bool	1	Status penandatanganan SPHP
sphp_exp_date	date	13	Tanggal kadaluarsa SPHP
application_status_code	int4	10	Status <i>application</i>
customer_id	int8	19	ID <i>user</i>
device_id	int4	10	ID Android <i>user</i>
twitter_username	varchar	50	Nama akun Twitter milik <i>user</i>
instagram_username	varchar	50	Nama akun Instagram milik <i>user</i>
loan_purpose_desc	text	100	Deskripsi tujuan peminjaman

Tabel 3.2 Struktur Tabel application (lanjutan)

Nama Kolom	Tipe Data	Ukuran	Keterangan
work_kodepos	varchar	100	Kode pos kantor <i>user</i>
job_function	varchar	100	Jabatan <i>user</i>
job_industry	varchar	100	Bidang pekerjaan <i>user</i>
college	varchar	100	Kampus tempat <i>user</i> kuliah
major	varchar	100	Jurusan <i>user</i> sewaktu kuliah
application_xid	int8	19	XID <i>application user</i>
app_version	varchar	10	Versi android yang digunakan
application_number	int4	10	Nomor aplikasi
gmail_scraped_status	varchar	15	Status apakah email <i>user</i> telah dilakukan <i>scrape</i>
payday	int4	10	Tanggal gaji <i>user</i>
product_line_code	int8	19	Kode produk yang dipilih <i>user</i>
partner_id	int4	10	ID <i>partner</i> JULO <i>user</i>
name_in_bank	varchar	100	Nama <i>user</i> pada rekening bank
customer_credit_limit_id	int4	10	ID batas kredit <i>user</i>
is_deleted	bool	1	Status <i>application</i> telah dihapus

b. Tabel loan

Gambaran struktur tabel loan berupa nama kolom dan tipe data terdapat pada Tabel 3.3, berfungsi untuk menyimpan data pinjaman *user*.

Tabel 3.3 Struktur Tabel loan

Nama Kolom	Tipe Data	Ukuran	Keterangan
loan_id	int4	10	Id data pinjaman <i>user</i>
customer_id	int4	10	Id data <i>user</i>
application_id	int4	10	Id aplikasi <i>user</i>
offer_id	int4	10	Id penawaran pinjaman <i>user</i>
loan_status_code	int4	10	Kode status pinjaman <i>user</i>
product_code	int4	10	Kode produk yang digunakan
application_xid	int4	10	Id aplikasi <i>user</i> untuk <i>partner</i>

Tabel 3.3 Struktur Tabel loan (lanjutan)

Nama Kolom	Tipe Data	Ukuran	Keterangan
agent_id	int4	10	Id agent yang melayani <i>user</i>
agent_2	int4	10	Id agent yang melayani <i>user</i>
agent_3	int4	10	Id agent yang melayani <i>user</i>
partner	int4	10	Id <i>partner user</i>
loan_amount	int8	19	Jumlah pinjaman <i>user</i>
loan_duration	int4	10	Durasi pinjaman <i>user</i>
sphp_sent_ts	timestamp	35	Waktu SPHP dikirimkan
sphp_accepted_ts	timestamp	35	Waktu SPHP diterima
first_installment_amount	int8	19	Jumlah <i>installment</i> pertama
installment_amount	int8	19	Total <i>installment</i>
cashback_earned_total	int8	19	Jumlah <i>cashback</i> yang diterima
initial_cashback	int8	19	Inisialisasi <i>cashback</i>
loan_disbursement_amount	int8	19	Jumlah dana yang dicairkan
loan_disbursement_method	varchar	100	Metode pencairan dana
fund_transfer_ts	timestamp	35	Waktu transfer dana
julo_bank_name	varchar	250	Nama bank JULO <i>user</i>
julo_bank_branch	varchar	100	Nama cabang JULO <i>user</i>
julo_bank_account_number	varchar	50	Nomor akun bank JULO
cycle_day	int4	10	<i>Cycle day</i>
cycle_day_change_date	date	13	Tanggal <i>cycle day</i>
cycle_day_requested	int4	10	Tanggal permintaan <i>cycle day</i>
is_ignore_calls	bool	1	Status penolakan panggilan
name_bank_validation_id	int8	19	Id validasi bank
disbursement_id	int8	19	Id pencairan dana <i>user</i>

c. Tabel payment

Gambaran struktur tabel payment berupa nama kolom dan tipe data terdapat pada Tabel 3.4, berfungsi untuk menyimpan data pembayaran *user*.

Tabel 3.4 Struktur Tabel payment

Nama Kolom	Tipe Data	Ukuran	Keterangan
payment_id	int4	10	Id <i>payment user</i>
loan_id	int4	10	Id pinjaman <i>user</i>

Tabel 3.4 Stuktur Tabel payment (lanjutan)

Nama Kolom	Tipe Data	Ukuran	Keterangan
payment_status_code	int4	10	Kode status pembayaran
payment_number	int4	10	Nomor pembayaran
due_date	timestamp	35	<i>Deadline</i> pembayaran
ptp_date	timestamp	35	Tanggal notifikasi
robocall_template_id	int4	10	Id <i>template</i> pesan robocall
ptp_robocall_phone_number	varchar	18	Nomor ponsel robocall
is_ptp_robocall_active	boolean	1	Status keaktifan nomor
due_amount	int8	19	Total pembayaran
installment_principal	int8	19	Jumlah <i>installment principal</i>
installment_interest	int8	19	Jumlah <i>installment interest</i>
paid_date	timestamp	35	Tanggal Pembayaran
paid_amount	int8	19	Total yang telah di bayar
redeemed_cashback	int8	19	Total <i>cashback</i> di <i>redeemed</i>
cashback_earned	int8	19	Total <i>cashback</i> diterima
late_fee_amount	int8	19	Biaya keterlambatan
late_fee_applied	int8	19	Biaya keterlambatan yang telah diberikan
discretionary_adjustment	int8	19	Pengaturan <i>discretionary</i>
is_robocall_active	boolean	1	Status keaktifan robocall
is_success_robocall	boolean	1	Status kesuksesan pesan
is_collection_called	boolean	1	Status panggilan <i>collection</i>
uncalled_date	date	13	Tanggal <i>uncalled</i>
reminder_call_date	timestamp	35	Waktu panggilan notifikasi
is_reminder_called	boolean	1	Status panggilan <i>reminder</i>
is_whatsapp	boolean	1	Status whatsapp
is_whatsapp_blasted	boolean	1	Status pengiriman whatsapp
paid_interest	int8	19	Total data <i>paid interest</i>
paid_principal	int8	19	Total data <i>paid principal</i>
paid_late_fee	int8	19	Total biaya keterlambatan
ptp_amount	int8	19	Total dana ptp
change_due_date_interest	int8	19	Total data <i>interest</i> pada <i>due date</i>
is_restructured	boolean	1	Status apakah <i>restructured</i>

d. Tabel repaymentTransaction

Gambaran struktur tabel repaymentTransaction berupa nama kolom dan tipe data terdapat pada Tabel 3.5, digunakan untuk menyimpan pengembalian dana.

Tabel 3.5 Struktur Tabel repaymentTransaction

Nama Kolom	Tipe Data	Ukuran	Keterangan
repayment_transaction_id	int4	10	Id transaksi pembayaran
partner_id	int4	10	Id <i>partner user</i>
customer_id	int4	10	Id <i>user</i>
loan_id	int4	10	Id pinjaman <i>user</i>
payment_id	int4	10	Id data pembayaran
payment_method_id	int4	10	Id metode pembayaran
event_date	date	13	Tanggal pembayaran
repayment_source	varchar	50	Sumber pembayaran
borrower_repaid	int8	19	Total pembayaran
borrower_repaid_principal	int8	19	Total <i>principal</i> pembayaran
borrower_repaid_interest	int8	19	Total <i>interest</i> pembayaran
borrower_repaid_late_fee	int8	19	Total biaya keterlambatan
lender_received	int8	19	Total dana diterima <i>partner</i>
lender_received_principal	int8	19	Total <i>principal</i> dana <i>partner</i>
lender_received_interest	int8	19	Total <i>interest</i> dana <i>partner</i>
lender_received_late_fee	int8	19	Total biaya keterlambatan
julo_fee_received	int8	19	Total bayaran pada JULO
julo_fee_received_principal	int8	19	Total data <i>principal</i> JULO
julo_fee_received_interest	int8	19	Total data <i>interest</i> JULO
julo_fee_received_late_fee	int8	19	Total biaya keterlambatan
due_amount_before	int8	19	Total tagihan sebelum pembayaran
due_amount_after	int8	19	Total tagihan sesudah
lender_balance_before	int8	19	Saldo <i>partner</i> sebelum
lender_balance_after	int8	19	Saldo <i>partner</i> sesudah

e. Tabel disbursement

Gambaran struktur tabel disbursement berupa nama kolom dan tipe data terdapat pada Tabel 3.6, digunakan untuk menyimpan data pencairan dana.

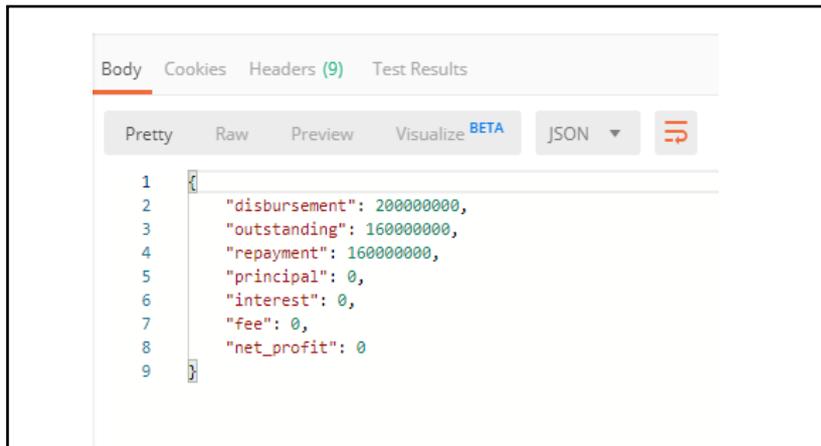
Tabel 3.6 Struktur Tabel disbursement

Nama Kolom	Tipe Data	Ukuran	Keterangan
disbursement_id	int4	10	Id pencairan dana <i>user</i>
loan_id	int4	10	Id pinjaman <i>user</i>
validation_status	varchar	50	Status validasi pencairan dana
validation_id	varchar	250	Id validasi pencairan dana
bank_code	varchar	50	Kode bank pencairan dana
bank_number	varchar	50	Nomor bank pencairan dana
disburse_status	varchar	50	Status pencairan dana
disburse_id	varchar	250	Id status pencairan dana
disburse_amout	int8	19	Jumlah dana yang dicairkan
external_id	int8	19	Id eksternal pencairan dana
retry_times	int4	10	Banyaknya pengulangan pencairan dana

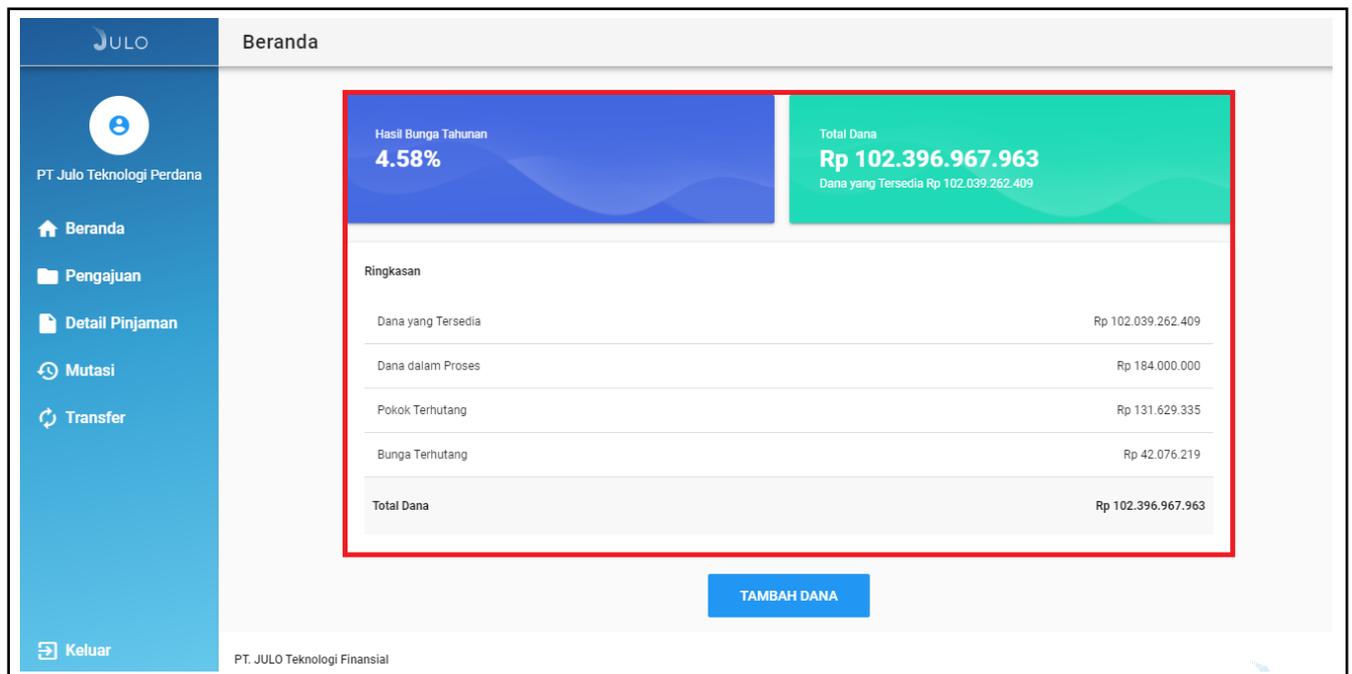
3.3.4 Implementasi

A. Summary View

Gambar 3.14 terdapat contoh hasil nilai pengembalian REST API Summary View. REST API tersebut mengembalikan data dalam bentuk JSON. Pengujian REST API tersebut dilakukan dengan aplikasi Postman. Data yang dikirimkan merupakan data yang didapat dari tabel payment dan tabel disbursement. Gambar 3.15 merupakan tampilan *frontend* dari penerapan API tersebut.



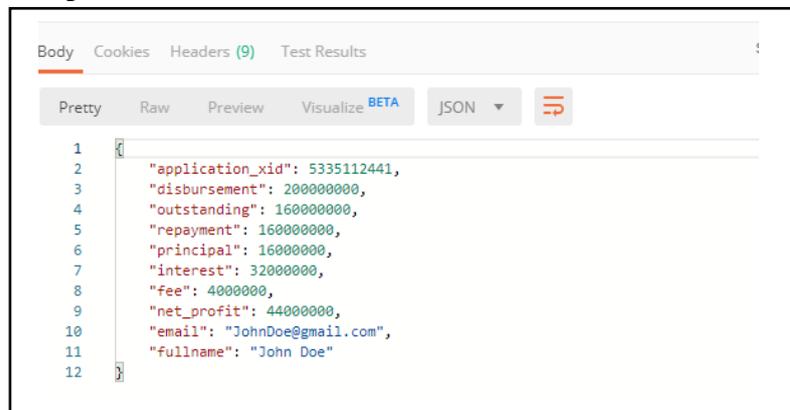
Gambar 3.14 JSON Summary View



Gambar 3.15 Tampilan Summary View

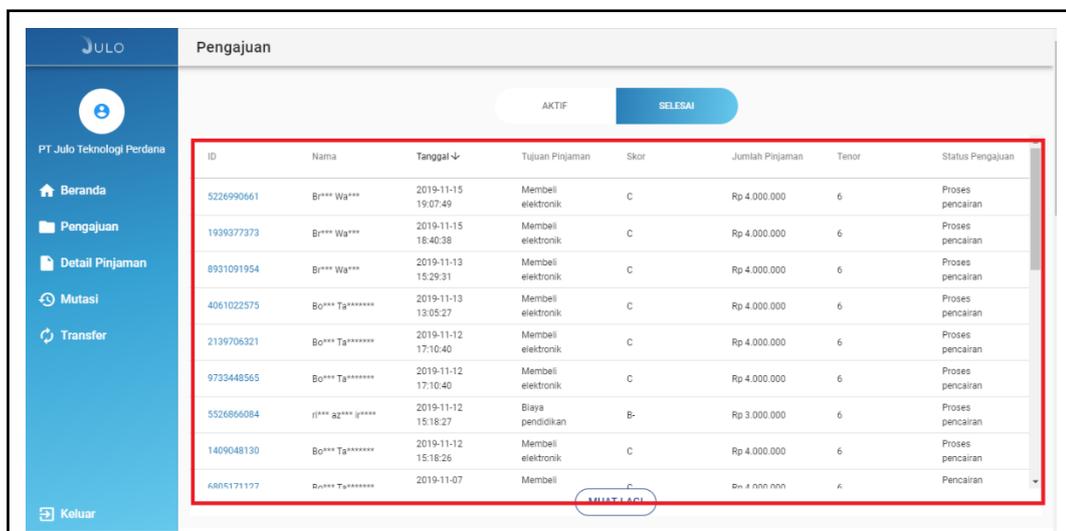
B. Report View

Gambar 3.16 terdapat contoh hasil nilai pengembalian REST API Report View. Gambar 3.17. Sama seperti Summary View, pengujian REST API tersebut dilakukan dengan aplikasi Postman. Data yang dikirimkan merupakan data perincian dari REST API Summary View yang merupakan perincian pinjaman pelanggan per aplikasi yang akan ditampilkan pada penyedia dana atau *partner*. Gambar 3.17 merupakan tampilan *frontend* pada Report View.



```
1 {
2   "application_xid": 5335112441,
3   "disbursement": 200000000,
4   "outstanding": 160000000,
5   "repayment": 160000000,
6   "principal": 160000000,
7   "interest": 32000000,
8   "fee": 4000000,
9   "net_profit": 44000000,
10  "email": "JohnDoe@gmail.com",
11  "fullname": "John Doe"
12 }
```

Gambar 3.16 JSON Report View

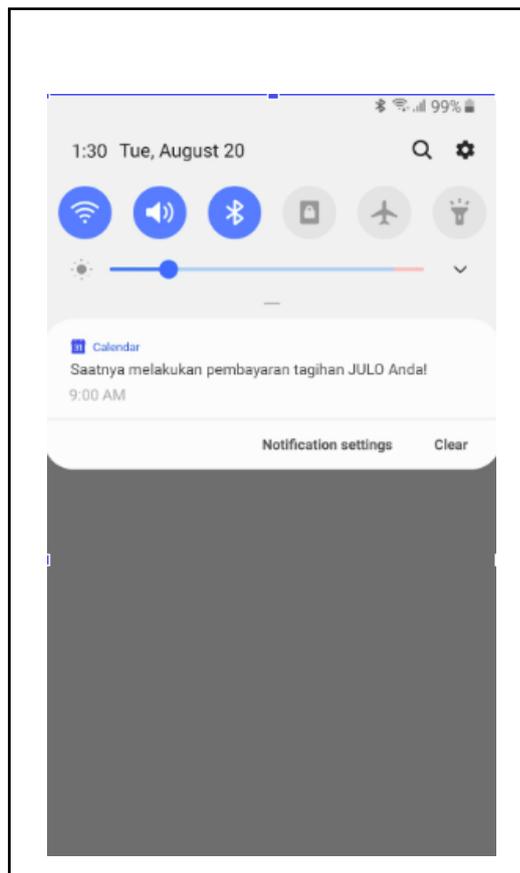


ID	Nama	Tanggal	Tujuan Pinjaman	Skor	Jumlah Pinjaman	Tenor	Status Pengajuan
5226990661	B**** Wa****	2019-11-15 19:07:49	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
1939277373	B**** Wa****	2019-11-15 18:40:38	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
8931091954	B**** Wa****	2019-11-13 15:29:31	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
4061022575	B**** Ta****	2019-11-13 13:05:27	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
2139706321	B**** Ta****	2019-11-12 17:10:40	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
973448565	B**** Ta****	2019-11-12 17:10:40	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
5526866084	l**** g****	2019-11-12 15:18:27	Biaya pendidikan	B-	Rp 3.000.000	6	Proses pencairan
1409048130	B**** Ta****	2019-11-12 15:18:26	Membeli elektronik	C	Rp 4.000.000	6	Proses pencairan
6805171177	B**** Ta****	2019-11-07	Membeli	C	Rp 4.000.000	6	Proses pencairan

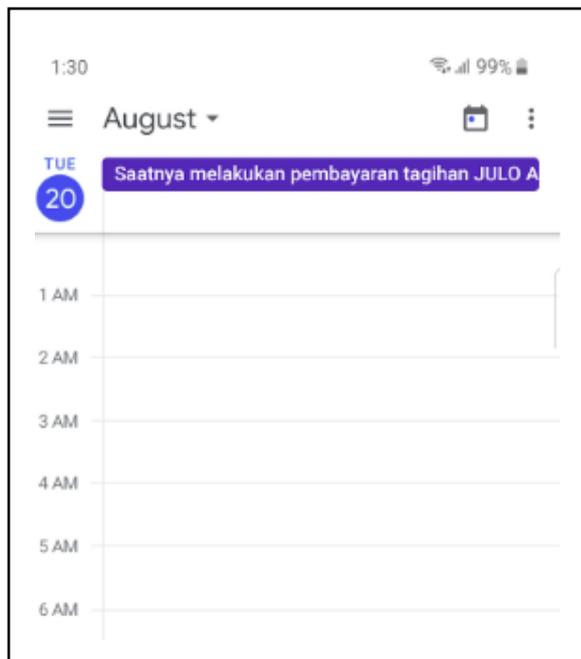
Gambar 3.17 Tampilan Report View

C. Notification Through Google Calendar

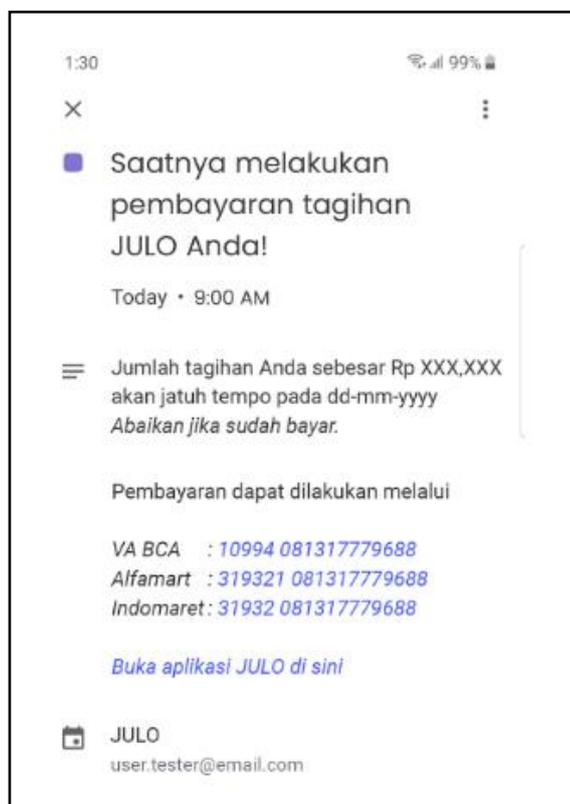
Pada gambar 3.18 terdapat contoh notifikasi dari Google Calendar pada perangkat *mobile* pelanggan. Isi dan perincian dari notifikasi Google Calendar terdapat pada Gambar 3.19 dan Gambar 3.20. Notifikasi Google Calendar tersebut akan otomatis muncul ketika satu hari sebelum tanggal batas pembayaran pinjaman. Pada isi dari Google Calendar terdapat rincian pinjaman pelanggan serta metode pembayarannya.



Gambar 3.18 Notifikasi Google Calendar



Gambar 3.19 Bagian dalam Google Calendar.



Gambar 3.20 Perincian Event Calendar.

D. Dockerize Juloserver

Proses untuk menjalankan sistem “MVP” juloserver pada Docker Container terdapat pada Gambar 3.21 hingga Gambar 3.25. Gambar tersebut menggambarkan tahapan Docker dalam menjalankan *project* dalam layanan docker container. Untuk menjalankan project pada docker container, dapat menggunakan perintah “docker-compose up”. Perintah tersebut hanya dapat dijalankan ketika Docker sudah terinstall pada perangkat.

```
File Edit View Search Terminal Help
+ juloserver git:(master) sudo docker-compose up
Swipe your finger across the fingerprint reader
Failed to match fingerprint
Swipe your finger across the fingerprint reader
Failed to match fingerprint
Swipe your finger across the fingerprint reader
Starting mvp_broker ...
Starting mvp_db ...
Starting mvp_broker ... done
Starting mvp_apl ...
Starting mvp_broker ... done
Recreating mvp_async ...
Recreating mvp_async ...
Recreating mvp_scheduler ...
Recreating mvp_scheduler ... done
Attaching to mvp_db, mvp_broker, mvp_apl, mvp_async, mvp_scheduler
mvp_1 | 2019-11-18 01:43:11.183 UTC [1] LOG: listening on IPv4 address '0.0.0.0', port 5432
mvp_1 | 2019-11-18 01:43:11.184 UTC [1] LOG: listening on IPv6 address '::', port 5432
mvp_1 | 2019-11-18 01:43:11.189 UTC [1] LOG: listening on Unix socket '/var/run/postgresql/.s.PGSQL.5432'
mvp_1 | 2019-11-18 01:43:11.206 UTC [18] LOG: database system was interrupted; last known up at 2019-09-25 11:11:58 UTC
mvp_1 | 2019-11-18 01:43:11.319 UTC [18] LOG: database system was not properly shut down; automatic recovery in progress
mvp_1 | 2019-11-18 01:43:11.382 UTC [18] LOG: redo starts at 0/5D7A7F8
mvp_1 | 2019-11-18 01:43:11.382 UTC [18] LOG: invalid record length at 0/5D7B88B: wanted 24, got 0
mvp_1 | 2019-11-18 01:43:11.382 UTC [18] LOG: redo done at 0/5D7B88B
mvp_1 | 2019-11-18 01:43:11.382 UTC [18] LOG: last completed transaction was at log time 2019-09-25 11:13:00.236559+00
mvp_1 | 2019-11-18 01:43:11.406 UTC [1] LOG: database system is ready to accept connections
mvp_1 | 2019-11-18 01:43:19.774 UTC [30] ERROR: column application.status_path locked does not exist at character 3417
mvp_1 | 2019-11-18 01:43:19.774 UTC [30] STATEMENT: SELECT application."cdater", application."udater", application."application_id",
application."customer_id", application."device_id", application."application", application."application_status_code", application."product_line_code", application."
partner_id", application."mantri_id", application."line_of_credit_id", application."loan_amount_request", application."validated_gr_cod
e", application."loan_duration_request", application."loan_purpose", application."loan_purpose_desc", application."marketing_source", ap
plication."payday", application."referral_code", application."is_own_phone", application."fullname", application."dob", application."b
irth_place", application."gender", application."ktp", application."address_street_num", application."address_province", application."ad
dress_kabupaten", application."address_kecamatan", application."address_kelurahan", application."address_kodepos", application."occupied_
since", application."home_status", application."landlord_mobile_phone", application."mobile_phone_1", application."new_mobile_phone", ap
plication."has_whatsapp_1", application."mobile_phone_2", application."has_whatsapp_2", application."email", application."bbn_pin", app
lication."twitter_username", application."instagram_username", application."marital_status", application."dependent", application."spous
```

Gambar 3.21 Proses Menjalankan Container Service database

```
File Edit View Search Terminal Help
fer_transaction."transfer_status" = "REQUESTED" GROUP BY "cashback_transfer_transaction"."application_id" subquery; args=( 'REQUESTED',
fer_transaction."transfer_status" = "PENDING" GROUP BY "cashback_transfer_transaction"."application_id" subquery; args=( 'PENDING',
fer_transaction."transfer_status" = "FAILED" GROUP BY "cashback_transfer_transaction"."application_id" subquery; args=( 'FAILED',
[2019-11-18 01:45:00,282] DEBUG/Worker-1 (0.000) SELECT COUNT(*) AS "count" FROM "loan" WHERE "loan"."loan_status_code" = 220;
args=(220,)
[2019-11-18 01:45:00,282] DEBUG/Worker-1 (0.000) SELECT COUNT(*) AS "count" FROM "loan" WHERE "loan"."loan_status_code" = 240;
args=(240,)
[2019-11-18 01:45:00,283] DEBUG/Worker-1 (0.000) SELECT COUNT(*) AS "count" FROM "loan" WHERE "loan"."loan_status_code" = 250;
args=(250,)
[2019-11-18 01:45:00,285] DEBUG/Worker-1 (0.000) SELECT COUNT(*) AS "count" FROM "loan" WHERE "loan"."cycle_day_requested" >= 1
; args=(1,)
[2019-11-18 01:45:00,288] DEBUG/Worker-1 (0.001) UPDATE "julo_dashboardbuckets" SET "app_120" = 0, "app_121" = 0, "app_122" = 0,
"app_123" = 0, "app_124" = 0, "app_125" = 0, "app_126" = 0, "app_127" = 0, "app_128" = 0, "app_129" = 0, "app_130" = 0, "app_131" = 0, "app_132" = 0, "app_133" = 0, "app_134" = 0, "app_135" = 0, "app_136" = 0, "app_137" = 0, "app_138" = 0, "app_139" = 0, "app_140" = 0, "app_141" = 0, "app_142" = 0, "app_143" = 0, "app_144" = 0, "app_145" = 0, "app_146" = 0, "app_147" = 0, "app_148" = 0, "app_149" = 0, "app_150" = 0, "app_151" = 0, "app_152" = 0, "app_153" = 0, "app_154" = 0, "app_155" = 0, "app_156" = 0, "app_157" = 0, "app_158" = 0, "app_159" = 0, "app_160" = 0, "app_161" = 0, "app_162" = 0, "app_163" = 0, "app_164" = 0, "app_165" = 0, "app_166" = 0, "app_167" = 0, "app_168" = 0, "app_169" = 0, "app_170" = 0, "app_171" = 0, "app_172" = 0, "app_173" = 0, "app_174" = 0, "app_175" = 0, "app_176" = 0, "app_177" = 0, "app_178" = 0, "app_179" = 0, "app_180" = 0, "app_181" = 0, "app_182" = 0, "app_183" = 0, "app_184" = 0, "app_185" = 0, "app_186" = 0, "app_187" = 0, "app_188" = 0, "app_189" = 0, "app_190" = 0, "app_191" = 0, "app_192" = 0, "app_193" = 0, "app_194" = 0, "app_195" = 0, "app_196" = 0, "app_197" = 0, "app_198" = 0, "app_199" = 0, "app_200" = 0, "app_201" = 0, "app_202" = 0, "app_203" = 0, "app_204" = 0, "app_205" = 0, "app_206" = 0, "app_207" = 0, "app_208" = 0, "app_209" = 0, "app_210" = 0, "app_211" = 0, "app_212" = 0, "app_213" = 0, "app_214" = 0, "app_215" = 0, "app_216" = 0, "app_217" = 0, "app_218" = 0, "app_219" = 0, "app_220" = 0, "app_221" = 0, "app_222" = 0, "app_223" = 0, "app_224" = 0, "app_225" = 0, "app_226" = 0, "app_227" = 0, "app_228" = 0, "app_229" = 0, "app_230" = 0, "app_231" = 0, "app_232" = 0, "app_233" = 0, "app_234" = 0, "app_235" = 0, "app_236" = 0, "app_237" = 0, "app_238" = 0, "app_239" = 0, "app_240" = 0, "app_241" = 0, "app_242" = 0, "app_243" = 0, "app_244" = 0, "app_245" = 0, "app_246" = 0, "app_247" = 0, "app_248" = 0, "app_249" = 0, "app_250" = 0, "app_251" = 0, "app_252" = 0, "app_253" = 0, "app_254" = 0, "app_255" = 0, "app_256" = 0, "app_257" = 0, "app_258" = 0, "app_259" = 0, "app_260" = 0, "app_261" = 0, "app_262" = 0, "app_263" = 0, "app_264" = 0, "app_265" = 0, "app_266" = 0, "app_267" = 0, "app_268" = 0, "app_269" = 0, "app_270" = 0, "app_271" = 0, "app_272" = 0, "app_273" = 0, "app_274" = 0, "app_275" = 0, "app_276" = 0, "app_277" = 0, "app_278" = 0, "app_279" = 0, "app_280" = 0, "app_281" = 0, "app_282" = 0, "app_283" = 0, "app_284" = 0, "app_285" = 0, "app_286" = 0, "app_287" = 0, "app_288" = 0, "app_289" = 0, "app_290" = 0, "app_291" = 0, "app_292" = 0, "app_293" = 0, "app_294" = 0, "app_295" = 0, "app_296" = 0, "app_297" = 0, "app_298" = 0, "app_299" = 0, "app_300" = 0, "app_301" = 0, "app_302" = 0, "app_303" = 0, "app_304" = 0, "app_305" = 0, "app_306" = 0, "app_307" = 0, "app_308" = 0, "app_309" = 0, "app_310" = 0, "app_311" = 0, "app_312" = 0, "app_313" = 0, "app_314" = 0, "app_315" = 0, "app_316" = 0, "app_317" = 0, "app_318" = 0, "app_319" = 0, "app_320" = 0, "app_321" = 0, "app_322" = 0, "app_323" = 0, "app_324" = 0, "app_325" = 0, "app_326" = 0, "app_327" = 0, "app_328" = 0, "app_329" = 0, "app_330" = 0, "app_331" = 0, "app_332" = 0, "app_333" = 0, "app_334" = 0, "app_335" = 0, "app_336" = 0, "app_337" = 0, "app_338" = 0, "app_339" = 0, "app_340" = 0, "app_341" = 0, "app_342" = 0, "app_343" = 0, "app_344" = 0, "app_345" = 0, "app_346" = 0, "app_347" = 0, "app_348" = 0, "app_349" = 0, "app_350" = 0, "app_351" = 0, "app_352" = 0, "app_353" = 0, "app_354" = 0, "app_355" = 0, "app_356" = 0, "app_357" = 0, "app_358" = 0, "app_359" = 0, "app_360" = 0, "app_361" = 0, "app_362" = 0, "app_363" = 0, "app_364" = 0, "app_365" = 0, "app_366" = 0, "app_367" = 0, "app_368" = 0, "app_369" = 0, "app_370" = 0, "app_371" = 0, "app_372" = 0, "app_373" = 0, "app_374" = 0, "app_375" = 0, "app_376" = 0, "app_377" = 0, "app_378" = 0, "app_379" = 0, "app_380" = 0, "app_381" = 0, "app_382" = 0, "app_383" = 0, "app_384" = 0, "app_385" = 0, "app_386" = 0, "app_387" = 0, "app_388" = 0, "app_389" = 0, "app_390" = 0, "app_391" = 0, "app_392" = 0, "app_393" = 0, "app_394" = 0, "app_395" = 0, "app_396" = 0, "app_397" = 0, "app_398" = 0, "app_399" = 0, "app_400" = 0, "app_401" = 0, "app_402" = 0, "app_403" = 0, "app_404" = 0, "app_405" = 0, "app_406" = 0, "app_407" = 0, "app_408" = 0, "app_409" = 0, "app_410" = 0, "app_411" = 0, "app_412" = 0, "app_413" = 0, "app_414" = 0, "app_415" = 0, "app_416" = 0, "app_417" = 0, "app_418" = 0, "app_419" = 0, "app_420" = 0, "app_421" = 0, "app_422" = 0, "app_423" = 0, "app_424" = 0, "app_425" = 0, "app_426" = 0, "app_427" = 0, "app_428" = 0, "app_429" = 0, "app_430" = 0, "app_431" = 0, "app_432" = 0, "app_433" = 0, "app_434" = 0, "app_435" = 0, "app_436" = 0, "app_437" = 0, "app_438" = 0, "app_439" = 0, "app_440" = 0, "app_441" = 0, "app_442" = 0, "app_443" = 0, "app_444" = 0, "app_445" = 0, "app_446" = 0, "app_447" = 0, "app_448" = 0, "app_449" = 0, "app_450" = 0, "app_451" = 0, "app_452" = 0, "app_453" = 0, "app_454" = 0, "app_455" = 0, "app_456" = 0, "app_457" = 0, "app_458" = 0, "app_459" = 0, "app_460" = 0, "app_461" = 0, "app_462" = 0, "app_463" = 0, "app_464" = 0, "app_465" = 0, "app_466" = 0, "app_467" = 0, "app_468" = 0, "app_469" = 0, "app_470" = 0, "app_471" = 0, "app_472" = 0, "app_473" = 0, "app_474" = 0, "app_475" = 0, "app_476" = 0, "app_477" = 0, "app_478" = 0, "app_479" = 0, "app_480" = 0, "app_481" = 0, "app_482" = 0, "app_483" = 0, "app_484" = 0, "app_485" = 0, "app_486" = 0, "app_487" = 0, "app_488" = 0, "app_489" = 0, "app_490" = 0, "app_491" = 0, "app_492" = 0, "app_493" = 0, "app_494" = 0, "app_495" = 0, "app_496" = 0, "app_497" = 0, "app_498" = 0, "app_499" = 0, "app_500" = 0, "app_501" = 0, "app_502" = 0, "app_503" = 0, "app_504" = 0, "app_505" = 0, "app_506" = 0, "app_507" = 0, "app_508" = 0, "app_509" = 0, "app_510" = 0, "app_511" = 0, "app_512" = 0, "app_513" = 0, "app_514" = 0, "app_515" = 0, "app_516" = 0, "app_517" = 0, "app_518" = 0, "app_519" = 0, "app_520" = 0, "app_521" = 0, "app_522" = 0, "app_523" = 0, "app_524" = 0, "app_525" = 0, "app_526" = 0, "app_527" = 0, "app_528" = 0, "app_529" = 0, "app_530" = 0, "app_531" = 0, "app_532" = 0, "app_533" = 0, "app_534" = 0, "app_535" = 0, "app_536" = 0, "app_537" = 0, "app_538" = 0, "app_539" = 0, "app_540" = 0, "app_541" = 0, "app_542" = 0, "app_543" = 0, "app_544" = 0, "app_545" = 0, "app_546" = 0, "app_547" = 0, "app_548" = 0, "app_549" = 0, "app_550" = 0, "app_551" = 0, "app_552" = 0, "app_553" = 0, "app_554" = 0, "app_555" = 0, "app_556" = 0, "app_557" = 0, "app_558" = 0, "app_559" = 0, "app_560" = 0, "app_561" = 0, "app_562" = 0, "app_563" = 0, "app_564" = 0, "app_565" = 0, "app_566" = 0, "app_567" = 0, "app_568" = 0, "app_569" = 0, "app_570" = 0, "app_571" = 0, "app_572" = 0, "app_573" = 0, "app_574" = 0, "app_575" = 0, "app_576" = 0, "app_577" = 0, "app_578" = 0, "app_579" = 0, "app_580" = 0, "app_581" = 0, "app_582" = 0, "app_583" = 0, "app_584" = 0, "app_585" = 0, "app_586" = 0, "app_587" = 0, "app_588" = 0, "app_589" = 0, "app_590" = 0, "app_591" = 0, "app_592" = 0, "app_593" = 0, "app_594" = 0, "app_595" = 0, "app_596" = 0, "app_597" = 0, "app_598" = 0, "app_599" = 0, "app_600" = 0, "app_601" = 0, "app_602" = 0, "app_603" = 0, "app_604" = 0, "app_605" = 0, "app_606" = 0, "app_607" = 0, "app_608" = 0, "app_609" = 0, "app_610" = 0, "app_611" = 0, "app_612" = 0, "app_613" = 0, "app_614" = 0, "app_615" = 0, "app_616" = 0, "app_617" = 0, "app_618" = 0, "app_619" = 0, "app_620" = 0, "app_621" = 0, "app_622" = 0, "app_623" = 0, "app_624" = 0, "app_625" = 0, "app_626" = 0, "app_627" = 0, "app_628" = 0, "app_629" = 0, "app_630" = 0, "app_631" = 0, "app_632" = 0, "app_633" = 0, "app_634" = 0, "app_635" = 0, "app_636" = 0, "app_637" = 0, "app_638" = 0, "app_639" = 0, "app_640" = 0, "app_641" = 0, "app_642" = 0, "app_643" = 0, "app_644" = 0, "app_645" = 0, "app_646" = 0, "app_647" = 0, "app_648" = 0, "app_649" = 0, "app_650" = 0, "app_651" = 0, "app_652" = 0, "app_653" = 0, "app_654" = 0, "app_655" = 0, "app_656" = 0, "app_657" = 0, "app_658" = 0, "app_659" = 0, "app_660" = 0, "app_661" = 0, "app_662" = 0, "app_663" = 0, "app_664" = 0, "app_665" = 0, "app_666" = 0, "app_667" = 0, "app_668" = 0, "app_669" = 0, "app_670" = 0, "app_671" = 0, "app_672" = 0, "app_673" = 0, "app_674" = 0, "app_675" = 0, "app_676" = 0, "app_677" = 0, "app_678" = 0, "app_679" = 0, "app_680" = 0, "app_681" = 0, "app_682" = 0, "app_683" = 0, "app_684" = 0, "app_685" = 0, "app_686" = 0, "app_687" = 0, "app_688" = 0, "app_689" = 0, "app_690" = 0, "app_691" = 0, "app_692" = 0, "app_693" = 0, "app_694" = 0, "app_695" = 0, "app_696" = 0, "app_697" = 0, "app_698" = 0, "app_699" = 0, "app_700" = 0, "app_701" = 0, "app_702" = 0, "app_703" = 0, "app_704" = 0, "app_705" = 0, "app_706" = 0, "app_707" = 0, "app_708" = 0, "app_709" = 0, "app_710" = 0, "app_711" = 0, "app_712" = 0, "app_713" = 0, "app_714" = 0, "app_715" = 0, "app_716" = 0, "app_717" = 0, "app_718" = 0, "app_719" = 0, "app_720" = 0, "app_721" = 0, "app_722" = 0, "app_723" = 0, "app_724" = 0, "app_725" = 0, "app_726" = 0, "app_727" = 0, "app_728" = 0, "app_729" = 0, "app_730" = 0, "app_731" = 0, "app_732" = 0, "app_733" = 0, "app_734" = 0, "app_735" = 0, "app_736" = 0, "app_737" = 0, "app_738" = 0, "app_739" = 0, "app_740" = 0, "app_741" = 0, "app_742" = 0, "app_743" = 0, "app_744" = 0, "app_745" = 0, "app_746" = 0, "app_747" = 0, "app_748" = 0, "app_749" = 0, "app_750" = 0, "app_751" = 0, "app_752" = 0, "app_753" = 0, "app_754" = 0, "app_755" = 0, "app_756" = 0, "app_757" = 0, "app_758" = 0, "app_759" = 0, "app_760" = 0, "app_761" = 0, "app_762" = 0, "app_763" = 0, "app_764" = 0, "app_765" = 0, "app_766" = 0, "app_767" = 0, "app_768" = 0, "app_769" = 0, "app_770" = 0, "app_771" = 0, "app_772" = 0, "app_773" = 0, "app_774" = 0, "app_775" = 0, "app_776" = 0, "app_777" = 0, "app_778" = 0, "app_779" = 0, "app_780" = 0, "app_781" = 0, "app_782" = 0, "app_783" = 0, "app_784" = 0, "app_785" = 0, "app_786" = 0, "app_787" = 0, "app_788" = 0, "app_789" = 0, "app_790" = 0, "app_791" = 0, "app_792" = 0, "app_793" = 0, "app_794" = 0, "app_795" = 0, "app_796" = 0, "app_797" = 0, "app_798" = 0, "app_799" = 0, "app_800" = 0, "app_801" = 0, "app_802" = 0, "app_803" = 0, "app_804" = 0, "app_805" = 0, "app_806" = 0, "app_807" = 0, "app_808" = 0, "app_809" = 0, "app_810" = 0, "app_811" = 0, "app_812" = 0, "app_813" = 0, "app_814" = 0, "app_815" = 0, "app_816" = 0, "app_817" = 0, "app_818" = 0, "app_819" = 0, "app_820" = 0, "app_821" = 0, "app_822" = 0, "app_823" = 0, "app_824" = 0, "app_825" = 0, "app_826" = 0, "app_827" = 0, "app_828" = 0, "app_829" = 0, "app_830" = 0, "app_831" = 0, "app_832" = 0, "app_833" = 0, "app_834" = 0, "app_835" = 0, "app_836" = 0, "app_837" = 0, "app_838" = 0, "app_839" = 0, "app_840" = 0, "app_841" = 0, "app_842" = 0, "app_843" = 0, "app_844" = 0, "app_845" = 0, "app_846" = 0, "app_847" = 0, "app_848" = 0, "app_849" = 0, "app_850" = 0, "app_851" = 0, "app_852" = 0, "app_853" = 0, "app_854" = 0, "app_855" = 0, "app_856" = 0, "app_857" = 0, "app_858" = 0, "app_859" = 0, "app_860" = 0, "app_861" = 0, "app_862" = 0, "app_863" = 0, "app_864" = 0, "app_865" = 0, "app_866" = 0, "app_867" = 0, "app_868" = 0, "app_869" = 0, "app_870" = 0, "app_871" = 0, "app_872" = 0, "app_873" = 0, "app_874" = 0, "app_875" = 0, "app_876" = 0, "app_877" = 0, "app_878" = 0, "app_879" = 0, "app_880" = 0, "app_881" = 0, "app_882" = 0, "app_883" = 0, "app_884" = 0, "app_885" = 0, "app_886" = 0, "app_887" = 0, "app_888" = 0, "app_889" = 0, "app_890" = 0, "app_891" = 0, "app_892" = 0, "app_893" = 0, "app_894" = 0, "app_895" = 0, "app_896" = 0, "app_897" = 0, "app_898" = 0, "app_899" = 0, "app_900" = 0, "app_901" = 0, "app_902" = 0, "app_903" = 0, "app_904" = 0, "app_905" = 0, "app_906" = 0, "app_907" = 0, "app_908" = 0, "app_909" = 0, "app_910" = 0, "app_911" = 0, "app_912" = 0, "app_913" = 0, "app_914" = 0, "app_915" = 0, "app_916" = 0, "app_917" = 0, "app_918" = 0, "app_919" = 0, "app_920" = 0, "app_921" = 0, "app_922" = 0, "app_923" = 0, "app_924" = 0, "app_925" = 0, "app_926" = 0, "app_927" = 0, "app_928" = 0, "app_929" = 0, "app_930" = 0, "app_931" = 0, "app_932" = 0, "app_933" = 0, "app_934" = 0, "app_935" = 0, "app_936" = 0, "app_937" = 0, "app_938" = 0, "app_939" = 0, "app_940" = 0, "app_941" = 0, "app_942" = 0, "app_943" = 0, "app_944" = 0, "app_945" = 0, "app_946" = 0, "app_947" = 0, "app_948" = 0, "app_949" = 0, "app_950" = 0, "app_951" = 0, "app_952" = 0, "app_953" = 0, "app_954" = 0, "app_955" = 0, "app_956" = 0, "app_957" = 0, "app_958" = 0, "app_959" = 0, "app_960" = 0, "app_961" = 0, "app_962" = 0, "app_963" = 0, "app_964" = 0, "app_965" = 0, "app_966" = 0, "app_967" = 0, "app_968" = 0, "app_969" = 0, "app_970" = 0, "app_971" = 0, "app_972" = 0, "app_973" = 0, "app_974" = 0, "app_975" = 0, "app_976" = 0, "app_977" = 0, "app_978" = 0, "app_979" = 0, "app_980" = 0, "app_981" = 0, "app_982" = 0, "app_983" = 0, "app_984" = 0, "app_985" = 0, "app_986" = 0, "app_987" = 0, "app_988" = 0, "app_989" = 0, "app_990" = 0, "app_991" = 0, "app_992" = 0, "app_993" = 0, "app_994" = 0, "app_995" = 0, "app_996" = 0, "app_997" = 0, "app_998" = 0, "app_999" = 0, "app_1000" = 0, "app_1001" = 0, "app_1002" = 0, "app_1003" = 0, "app_1004" = 0, "app_1005" = 0, "app_1006" = 0, "app_1007" = 0, "app_1008" = 0, "app_1009" = 0, "app_1010" = 0, "app_1011" = 0, "app_1012" = 0, "app_1013" = 0, "app_1014" = 0, "app_1015" = 0, "app_1016" = 0, "app_1017" = 0, "app_1018" = 0, "app_1019" = 0, "app_1020" = 0, "app_1021" = 0, "app_1022" = 0, "app_1023" = 0, "app_1024" = 0, "app_1025" = 0, "app_1026" = 0, "app_1027" = 0, "app_1028" = 0, "app_1029" = 0, "app_1030" = 0, "app_1031" = 0, "app_1032" = 0, "app_1033" = 0, "app_1034" = 0, "app_1035" = 0, "app_1036" = 0, "app_1037" = 0, "app_1038" = 0, "app_1039" = 0, "app_1040" = 0, "app_1041" = 0, "app_1042" = 0, "app_1043" = 0, "app_1044" = 0, "app_1045" = 0, "app_1046" = 0, "app_1047" = 0, "app_1048" = 0, "app_1049" = 0, "app_1050" = 0, "app_1051" = 0, "app_1052" = 0, "app_1053" = 0, "app_1054" = 0, "app_1055" = 0, "app_1056" = 0, "app_1057" = 0, "app_1058" = 0, "app_1059" = 0, "app_1060" = 0, "app_1061" = 0, "app_1062" = 0, "app_1063" = 0, "app_1064" = 0, "app_1065" = 0, "app_1066" = 0, "app_1067" = 0, "app_1068
```

```

File Edit View Search Terminal Help
broker_1 | client unexpectedly closed TCP connection
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:14 ===
broker_1 | Starting RabbitMQ 3.6.15 on Erlang 19.3
broker_1 | Copyright (C) 2007-2018 Pivotal Software, Inc.
broker_1 | Licensed under the MPL. See http://www.rabbitmq.com/
broker_1 |
broker_1 |           RabbitMQ 3.6.15. Copyright (C) 2007-2018 Pivotal Software, Inc.
broker_1 |           Licensed under the MPL. See http://www.rabbitmq.com/
broker_1 | ##### Logs: tty
broker_1 | #####      tty
broker_1 | #####
broker_1 |           Starting broker...
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:14 ===
broker_1 | node       : rabbit@062ca2652153
broker_1 | home dir  : /var/lib/rabbitmq
broker_1 | config file(s) : /etc/rabbitmq/rabbitmq.conf
broker_1 | cookie hash : ejXl9eqw2/ESFTBXngcg==
broker_1 | log       : tty
broker_1 | sasl log  : tty
broker_1 | database dir : /var/lib/rabbitmq/mnesia/rabbit@062ca2652153
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:17 ===
broker_1 | Memory high watermark set to 3149 MiB (3301995315 bytes) of 7872 MiB (8254988288 bytes) total
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:17 ===
broker_1 | Enabling free disk space monitoring
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:17 ===
broker_1 | Disk free limit set to 50MB
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:17 ===
broker_1 | Limiting to approx 1048476 file handles (943626 sockets)
broker_1 |
broker_1 | =INFO REPORT=== 18-Nov-2019:01:43:17 ===
broker_1 | FHC read buffering: OFF
broker_1 | FHC write buffering: ON

```

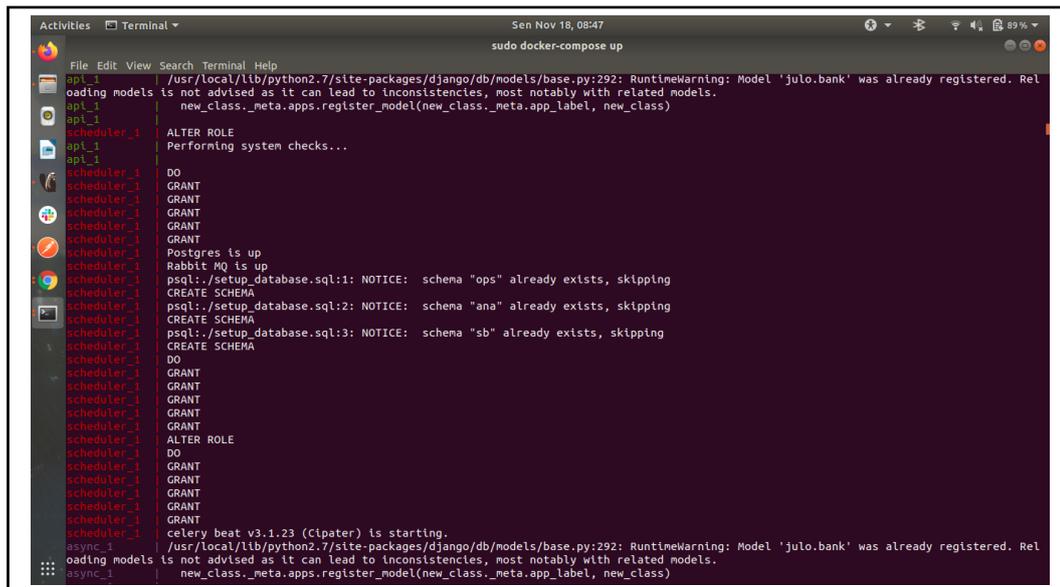
Gambar 3.23 Proses Menjalankan Container Service broker

```

File Edit View Search Terminal Help
broker_1 | client unexpectedly closed TCP connection
broker_1 |
broker_1 | =WARNING REPORT=== 18-Nov-2019:01:43:21 ===
broker_1 | closing AMQP connection <0.487.0> (172.18.0.4:40820 -> 172.18.0.5:5672, vhost: '/', user: 'guest'):
broker_1 | client unexpectedly closed TCP connection
broker_1 |
broker_1 | CREATE SCHEMA
broker_1 | psql:/setup_database.sql:1: NOTICE: schema "ops" already exists, skipping
broker_1 | psql:/setup_database.sql:2: NOTICE: schema "ana" already exists, skipping
broker_1 | psql:/setup_database.sql:1: NOTICE: schema "ops" already exists, skipping
broker_1 | CREATE SCHEMA
broker_1 | CREATE SCHEMA
broker_1 | psql:/setup_database.sql:2: NOTICE: schema "ana" already exists, skipping
broker_1 | psql:/setup_database.sql:3: NOTICE: schema "sb" already exists, skipping
broker_1 | DO
broker_1 | GRANT
broker_1 | ALTER ROLE
broker_1 | DO
broker_1 | GRANT
broker_1 | PostgreSQL is up
broker_1 | RabbitMQ is up
broker_1 | psql:/setup_database.sql:3: NOTICE: schema "sb" already exists, skipping
broker_1 | psql:/setup_database.sql:1: NOTICE: schema "ops" already exists, skipping
broker_1 | CREATE SCHEMA
broker_1 | CREATE SCHEMA
broker_1 | CREATE SCHEMA
broker_1 | psql:/setup_database.sql:2: NOTICE: schema "ana" already exists, skipping
broker_1 | DO

```

Gambar 3.24 Proses Menjalankan Container Service api



```
Sen Nov 18, 08:47
sudo docker-compose up
api_1 /usr/local/lib/python2.7/site-packages/django/db/models/base.py:292: RuntimeWarning: Model 'julo.bank' was already registered. Rel
oadng models is not advlseed as it can lead to inconsistencies, most notably with related models.
api_1 new_class._meta.apps.register_model(new_class._meta.app_label, new_class)
scheduler_1 ALTER ROLE
api_1 Performing system checks...
api_1
scheduler_1 DO
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 Postgrfas is up
scheduler_1 Rabbit MQ is up
scheduler_1 psql:./setup_database.sql:1: NOTICE: schema "ops" already exists, skipping
scheduler_1 CREATE SCHEMA
scheduler_1 psql:./setup_database.sql:2: NOTICE: schema "ana" already exists, skipping
scheduler_1 CREATE SCHEMA
scheduler_1 psql:./setup_database.sql:3: NOTICE: schema "sb" already exists, skipping
scheduler_1 CREATE SCHEMA
scheduler_1 DO
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 ALTER ROLE
scheduler_1 DO
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 GRANT
scheduler_1 celery beat v3.1.23 (Clpater) is starting.
api_1 /usr/local/lib/python2.7/site-packages/django/db/models/base.py:292: RuntimeWarning: Model 'julo.bank' was already registered. Rel
oadng models is not advlseed as it can lead to inconsistencies, most notably with related models.
async_1 new_class._meta.apps.register_model(new_class._meta.app_label, new_class)
```

Gambar 3.25 Proses Menjalankan Container Service scheduler

3.3.5 Kendala yang Ditemukan

Pada saat pengerjaan tugas-tugas yang diberikan, terdapat beberapa kendala yang ditemukan, diantaranya adalah sebagai berikut :

- Proses *codereview* pada *pullrequest* github yang bergantung pada *engineer* lainnya, serta proses perbaikan penulisan dan logika program membutuhkan waktu yang cukup lama.
- Proses beradaptasi terhadap kerangka kerja baru, dan proses pembelajaran terhadap program yang ingin dikembangkan dari *engineer* lain membutuhkan waktu yang cukup lama.

3.3.6 Solusi Atas Kendala yang Ditemukan

Solusi untuk menangani kendala-kendala yang telah disebutkan pada subbab sebelumnya adalah sebagai berikut :

- Bersifat inisiatif pada saat ingin meminta *codereview* pada *engineer* lainnya dengan berkomunikasi secara langsung, serta melakukan *codereview* sendiri secara individu agar tidak banyak potongan kode yang harus diperbaiki.
- Bersifat inisiatif untuk mempelajari hal-hal baru seperti kerangka kerja baru dan alur program *engineer* lain yang ingin dikembangkan.