Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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.

Perpustakaan Univeristas Multimedia Nusantara

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di PT. JULO Teknologi Finansial, penulis memegang jabatan sebagai QA Automation Intern di divisi *Engineering*, dengan supervisi dari Bapak Rayhan Dwi Ananda, selaku *Software* QA di bidang *automation* PT. JULO Teknologi Finansial. Bapak Rayhan berperan dalam memberikan informasi dan bimbingan dalam mengembangkan *script automation testing* yang diterapkan pada *regression test* aplikasi JULO.

3.2 Tugas yang Dilakukan

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

- Mengembangkan, *framework automation testing* yang digunakan untuk melakukan *regression* test pada aplikasi JULO.
- Mengimplementasikan *automated regression test* yang telah dikembangkan pada setiap perilisan versi aplikasi terbaru.
- Mendokumentasikan dan menjaga *log* yang kuat dari semua kasus uji dan hasil uji.

- berkolaborasi dengan anggota tim teknis lainnya untuk menemukan metode terbaik untuk menyelesaikan masalah dalam aplikasi dan sistem.

3.3 Uraian Pelaksanaan

3.3.1 Proses Pelaksanaan

Kerja magang dilaksanakan selama 3 (tiga) bulan atau 12 minggu dengan *timeline* kerja sebagai berikut.

Tabel 3.1 Jadwal Kerja Magang

Nama Kegiatan	Minggu													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pengenalan *framework* dan alur kerja														
Manual testing														
Integrasi *auto test mobile* dan *web app*														
Membuat *testsuite* STL														
Improve handler untuk *release* terbaru														
Auto regression test														
Sustaining														
Bug fix testing														

3.3.2 Framework dan Alur Kerja yang Digunakan

Sepanjang minggu pertama magang, belum ada tugas yang diberikan oleh *supervisor*. Dalam minggu tersebut, lebih diutamakan untuk beradaptasi dan mengenal *framework* dan bahasa pemrograman yang digunakan di PT. JULO Teknologi Finansial. *Framework* yang digunakan untuk *automation testing* di PT. JULO Teknologi Finansial adalah Robot Framework dan dikombinasikan dengan bahasa pemrograman Python.

Robot Framework adalah *framework open source* yang biasa digunakan untuk *acceptance testing, acceptance test driven development* (AATD), dan *robotic process automation* (RPA). *Framework* ini dinilai lebih mudah karena menggunakan *syntax* yang mudah dipahami dan menggunakan pendekatan dengan kata kunci *(keyword)*. Kemampuan *testing* dari Robot Framework dapat diimplementasikan dengan Python atau Java, dan pengguna dapat mengembangkan *keyword* dari *keyword* yang telah ada sebelumnya menggunakan *syntax* yang sama untuk membuat *test case* (Pekka Klarck, 2008, robotframework.org, 18 September 2019).

Robot Framework sendiri merupakan aplikasi *open source* yang dirilis di bawah Apache Licence 2.0. *Library* dan *tools* yang berada dalam *framework* ini juga merupakan *open source software*. Pertama kali, Robot Framework dikembangkan di Nokia Networks dan sekarang disponsori oleh Robot Framework Foundation.

Di PT. JULO Teknologi Finansial, Robot Framework digunakan untuk keperluan *regression testing*. Ketika ada rilis terbaru dari *developer*, maka perlu

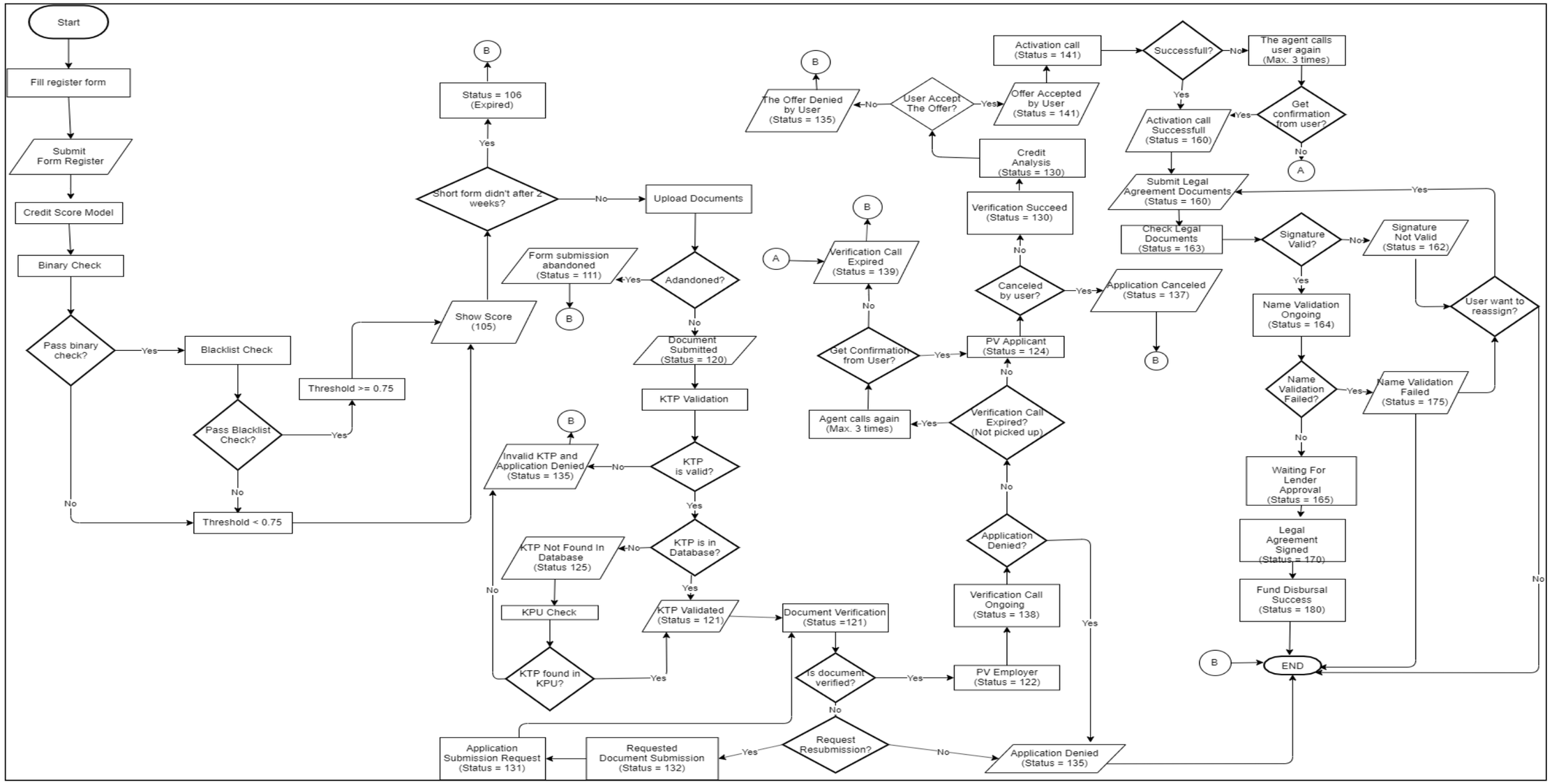
dilakukan *regression test* untuk memastikan fitur-fitur lain tidak terpengaruh atau terganggu oleh fitur baru yang dirilis. Karena proses *regression test* sangat panjang dan memerlukan waktu yang lama bila dilakukan secara *manual*, digunakanlah Robot Framework untuk mengubah proses *testing* yang sebelumnya dilakukan oleh manusia, menjadi proses yang bisa berjalan otomatis tanpa perlu intervensi manusia.

Adapun proses *automated regression test* yang dijalankan harus sesuai dengan *flowchart* yang telah ditentukan oleh perusahaan. Gambar 3.1 merupakan gambar *flowchart disbursement* yang telah dilakukan *update* pada bulan September 2019 ini. Apabila aplikasi yang diuji tidak sesuai dengan alur yang ditentukan, maka akan dianggap gagal dan *developer* wajib melakukan *code fixing.*

Di PT. JULO Teknologi Finansial, *database* yang dipakai menggunakan PostgreSQL yang dapat diakses melalui DBeaver. Penulisan *syntax* pada PostgreSQL kurang lebih sama dengan *syntax* MYSQL pada umumnya. Dalam laporan magang ini, tidak semua tabel *database* akan terlibat dalam *test case*. Hanya ada beberapa tabel yang digunakan untuk keperluan *automation testing.*

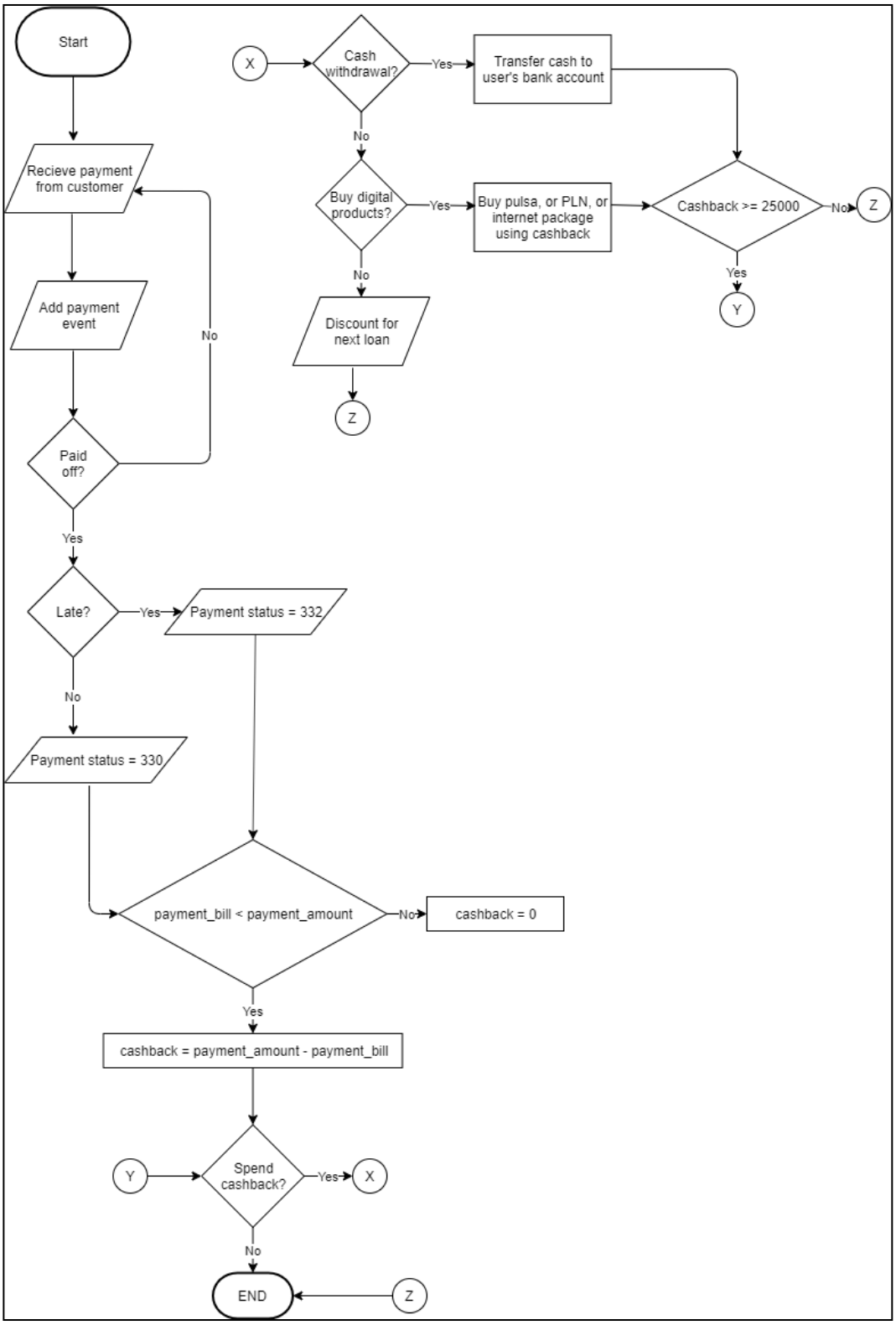
3.3.3 Perancangan Sistem

A. Flowchart

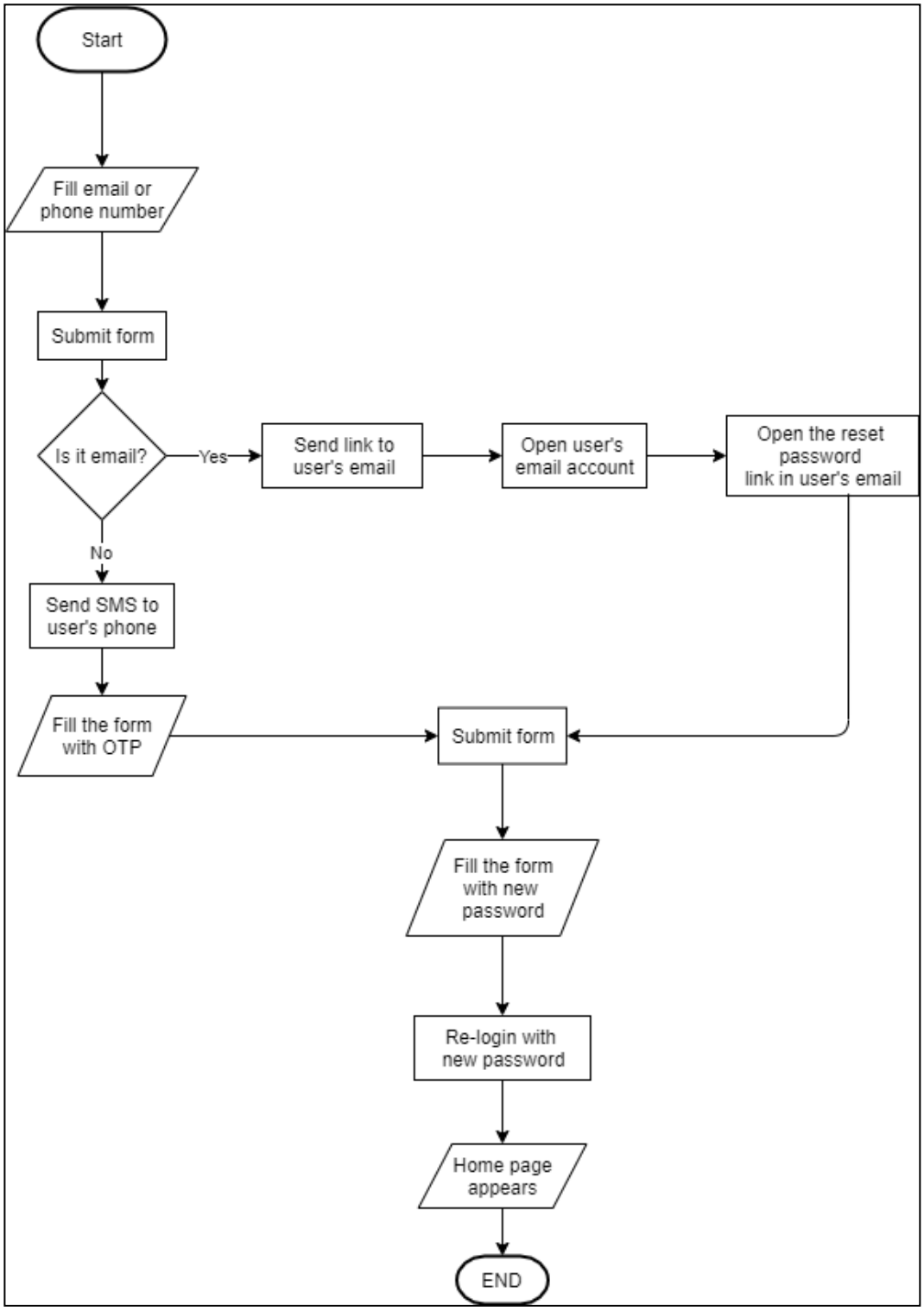


Gambar 3.1 Flowchart *disbursement* JULO

Sumber: Dokumentasi Internal Perusahaan

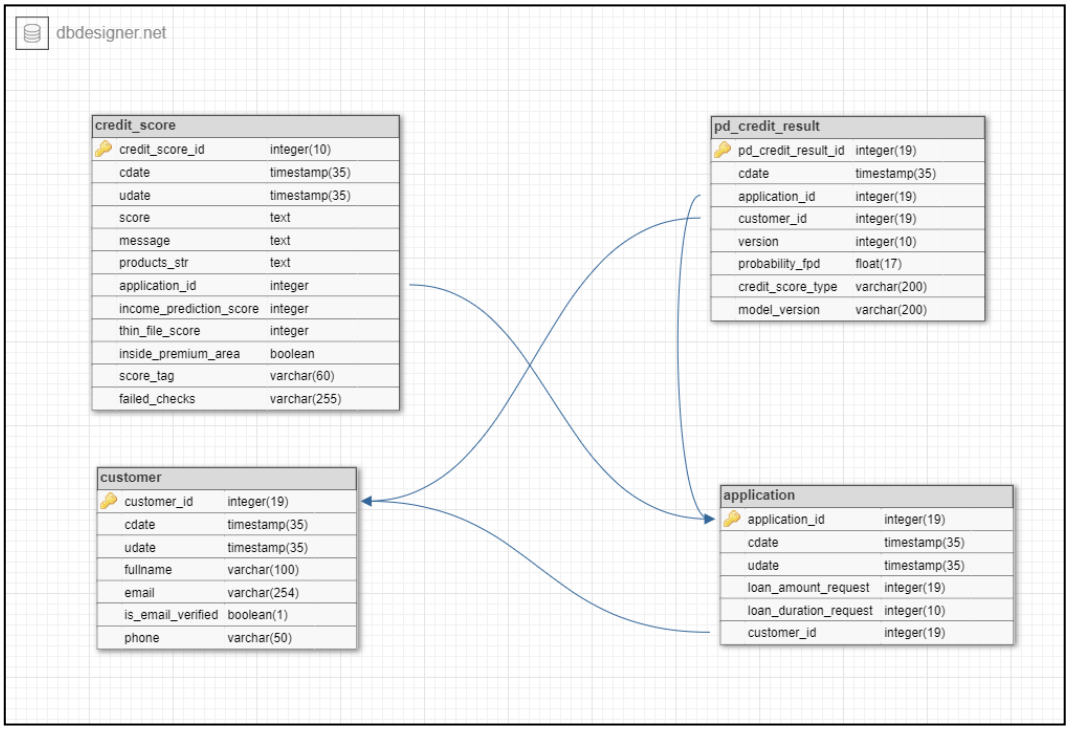


Gambar 3.2 Flowchart *payment* dan *cashback*



Gambar 3.3 Flowchart *forgot or reset password*

B. Relasi Antar Tabel



Gambar 3.4 Relasi Antar Tabel

Pada Gambar 3.4, terdapat 4 tabel utama dalam *database* yang digunakan untuk keperluan *testing* aplikasi. Setiap data yang dikirim dari proses registrasi *user* akan dikirim ke tabel application dan tabel customer. Hasil pengolahan data yang dilakukan oleh *machine learning* akan disimpan dalam tabel credit_score dan pd_credit_result.

C. Struktur Tabel

Database yang digunakan adalah PostgreSQL. Berikut adalah struktur tabel-tabel yang digunakan.

Tabel : credit_score

Fungsi : untuk menyimpan data skor kredit

Tabel 3.2 Tabel credit_score

Nama Kolom	Tipe Data	Panjang	Keterangan
cdate	timestamp	35	*Created Date*
udate	timestamp	35	*Updated Date*
credit_score_id **(PK)**	int4	10	ID *credit score*
score	text		Skor yang ditampilkan
message	text		Deskripsi terkait skor yang didapatkan
products_str	text		Produk yang dapat dipilih
application_id	int4	10	ID aplikasi
income_prediction_score	numeric	15	Skor prediksi pendapatan *user*
thin_file_score	numeric	15	Skor file
inside_premium_area	bool	1	*Boolean check* lokasi user
score_tag	varchar	60	Tag skor
credit_limit	int8	19	Limit pinjaman
failed_checks	varchar	255	Keterangan kegagalan lolos *binary check*

Tabel : pd_credit_result

Fungsi : untuk menyimpan probabilitas skor kredit

Tabel 3.3 Tabel pd_credit_result

Nama Kolom	Tipe Data	Panjang	Keterangan
pd_credit_model_result_id **(PK)**	int8	19	ID *credit model* yang dihasilkan
cdate	timestamp	35	*Created Date*
application_id	int8	19	ID aplikasi
customer_id	int8	19	ID *user*
version	int4	10	Versi rilis
probability_fpd	float8	17	Probabilitas skor
product	varchar	20	Produk yang bisa dipilih *user*
credit_score_type	varchar	200	Tipe skor yang didapatkan
model_version	varchar	200	Versi model data

Tabel : application

Fungsi : untuk menyimpan data *user* saat mengajukan pinjaman

Tabel 3.4 Tabel application

Nama Kolom	Tipe Data	Panjang	Keterangan
cdate	timestamp	35	*Created Date*
udate	timestamp	35	*Updated Date*
application_id	int8	19	ID aplikasi
loan_amount_request	int8	19	Jumlah pengajuan pinjaman
loan_duration_request	int4	10	Durasi peminjaman
loan_purpose	varchar	100	Tujuan peminjaman
marketing_source	varchar	100	Sumber darimana *user* mengetahui JULO
referral_code	varchar	20	Kode *referral*
is_own_phone	bool	1	Apakah ponsel yang digunakan milik *user* sendiri atau bukan
fullname	varchar	100	Nama lengkap *user*
dob	date	13	Tanggal lahir *user*
gender	varchar	10	Jenis kelamin
ktp	varchar	16	NIK
address_street_num	varchar	100	Nomor jalan
address_provinsi	varchar	100	Letak Provinsi
address_kabupaten	varchar	100	Letak Kabupaten atau kota
address_kecamatan	varchar	100	Letak Kecamatan
address_kelurahan	varchar	100	Alamat Kelurahan
address_kodepos	varchar	5	Kode pos
occupied_since	date	13	Tanggal *user* menempati alamat yang dimaksud
home_status	varchar	50	Status kepemilikan bangunan

Tabel 3.4 Tabel application (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
landlord_mobile_phone	varchar	50	Nomor telepon rumah
mobile_phone_1	varchar	50	Nomor HP utama
has_whatsapp_1	bool	1	Nilai *boolean* jika nomor HP utama memiliki akun Whatsapp
mobile_phone_2	varchar	50	Nomor HP lainnya
has_whatsapp_2	bool	1	Nilai *boolean* jika nomor HP memiliki akun Whatsapp
email	varchar	254	Email *user*
bbm_pin	varchar	50	Pin BBM
marital_status	varchar	50	Status pernikahan *user*
dependent	int4	10	Tanggungan
spouse_name	varchar	100	Nama orang tua *user*
spouse_dob	date	13	Tanggal lahir orang tua *user*
spouse_mobile_phone	varchar	50	Nomor HP orang tua *user*
spouse_has_whatsapp	bool	1	Nilai *boolean* apakah nomor HP orang tua *user* memiliki akun Whatsapp
kin_name	varchar	100	Nama saudara kandung *user*
kin_dob	date	13	Tanggal lahir saudara kandung *user*
kin_gender	varchar	10	Jenis kelamin saudara kandung *user*

Tabel 3.4 Tabel application (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
kin_mobile_phone	varchar	50	Nomor HP saudara kandung *user*
kin_relationship	varchar	50	Jenis hubungan saudara kandung dengan *user*
job_type	varchar	50	Jenis pekerjaan yang dimiliki *user*
job_description	varchar	100	Deskripsi pekerjaan yang dimiliki *user*
company_name	varchar	100	Nama perusahaan di mana *user* bekerja
company_phone_number	varchar	50	Nomor telepon perusahaan
job_start	date	13	Tanggal *user* mulai bekerja
monthly_income	int8	19	Total pendapatan bulanan
income_1	int8	19	Detail pendapatan dari sumber yang bersangkutan
income_2	int8	19	Detail pendapatan dari sumber yang bersangkutan
income_3	int8	19	Detail pendapatan dari sumber yang bersangkutan
last_education	varchar	50	Pendidikan terakhir *user*
graduation_year	int4	10	Tahun lulus *user*
gpa	float8	17	IPK *user* (jika S1 ke atas)
has_other_income	bool	1	Nilai *boolean* apakah *user* memiliki pendapatan sampingan

Tabel 3.4 Tabel application (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
other_income_amount	int8	19	Jumlah pendapatan sampingan
other_income_source	varchar	250	Keterangan jika *user* memiliki sumber pendapatan lain
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 yang dimiliki *user*
vehicle_ownership_1	varchar	50	Status kepemilikan kendaraan
bank_name	varchar	250	Nama bank yang digunakan *user*
bank_branch	varchar	100	Cabang bank
bank_account_number	varchar	50	Nomor rekening *user*
is_term_accepted	bool	1	Nilai *boolean* apakah *user* menyetujui form persetujuan
is_term_verification_agreed	bool	1	Nilai *boolean* apakah *user* menyetujui form verfikasi data
is_document_submitted	bool	1	Nilai *boolean* apakah dokumen telah disubmit
is_sphp_signed	bool	1	Nilai *boolean* apakah SPHP telah ditandatangani *user*
sphp_exp_date	date	13	Tanggal kadaluarsa SPHP
application_status_code	int4	10	Status *application*
customer_id	int8	19	ID *user*

Tabel 3.4 Tabel application (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
device_id	int4	10	ID Android *user*
twitter_username	varchar	50	Nama akun Twitter milik *user*
instagram_username	varchar	50	Nama akun Instagram milik *user*
loan_purpose_desc	text		Deskripsi tujuan peminjaman
work_kodepos	varchar	100	Kode pos kantor di mana *user* bekerja
job_function	varchar	100	Jabatan *user*
job_industry	varchar	100	Bidang pekerjaan *user*
college	varchar	100	Kampus tempat *user* kuliah (jika S1 ke atas)
major	varchar	100	Jurusan *user* sewaktu kuliah
application_xid	int8	19	XID *application user*
app_version	varchar	10	Versi aplikasi android yang digunakan
application_number	int4	10	Nomor aplikasi
gmail_scraped_status	varchar	15	Status apakah email *user* telah dilakukan *scrape*
payday	int4	10	Tanggal gajian *user*
product_line_cone	int8	19	Kode produk yang dipilih *user*
partner_id	int4	10	ID *partner* JULO yang digunakan user
name_in_bank	varchar	100	Nama rekening *user* yang terdaftar di bank

Tabel 3.4 Tabel application (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
customer_credit_limit_id	int4	10	ID batas kredit *user*
is_deleted	bool	1	Nilai *boolean* apakah *application* telah dihapus

Tabel : customer

Fungsi : untuk menyimpan data pribadi *user* atau *customer*

Tabel 3.5 Tabel customer

Nama Kolom	Tipe Data	Panjang	Keterangan
cdate	timestamp	35	Waktu ter-*generate*
udate	timestamp	35	Waktu *update*
customer_id	int8	19	ID *user*
fullname	varchar	100	Nama lengkap *user*
email	varchar	254	Email *user*
is_email_verified	bool	1	Pengecekan verifikasi email
phone	varchar	50	Nomor HP *user*
country	varchar	50	Negara tempat tinggal *user*
self_referral_code	varchar	20	Kode *referral user*
email_verification_key	varchar	50	*Key* untuk *user* melakukan verifikasi *email*
email_key_exp_date	timestamp	35	Waktu kadaluarsa *key* verifikasi *email*
reset_password_key	varchar	50	*Key* untuk reset *password*
reset_password_exp_date	timestamp	35	Waktu kadaluarsa *key* untuk reset *password*

Tabel 3.5 Tabel customer (lanjutan)

Nama Kolom	Tipe Data	Panjang	Keterangan
auth_user_id	int4	10	ID autentikasi *user*
can_reapply	bool	1	Pengecekan apakah *user* bisa *reapply*
is_review_submitted	bool	1	Pengecekan apakah form sudah di-*review*
nik	varchar	16	NIK *user*
appsflyer_device_id	varchar	50	*Device* ID *user*
advertising_id	varchar	50	Sumber marketing *user*
disabled_reapply_date	timestamp	35	Tanggal *user* tidak bisa *reapply*
can_reapply_date	timestamp	35	Tanggal *user reapply*
potential_skip_pv_dv	bool	1	Pengecekan *special user*
google_access_token	varchar	500	Google token untuk register
google_refresh_token	varchar	500	Google token untuk *refresh*
is_digisign_activated	bool	1	Pengecekan *digital signature* apakah aktif atau tidak
is_digisign_registered	bool	1	Pengecekan apakah *digital signature* terdaftar
customer_xid	int8	19	ID *expired user*
dob	date	13	Tanggal lahir
gender	varchar	10	Jenis kelamin
can_notify	bool	1	Pengecekan apakah *user* dapat dinotifikasi

Tabel 3.5 Tabel customer (lanjutan)

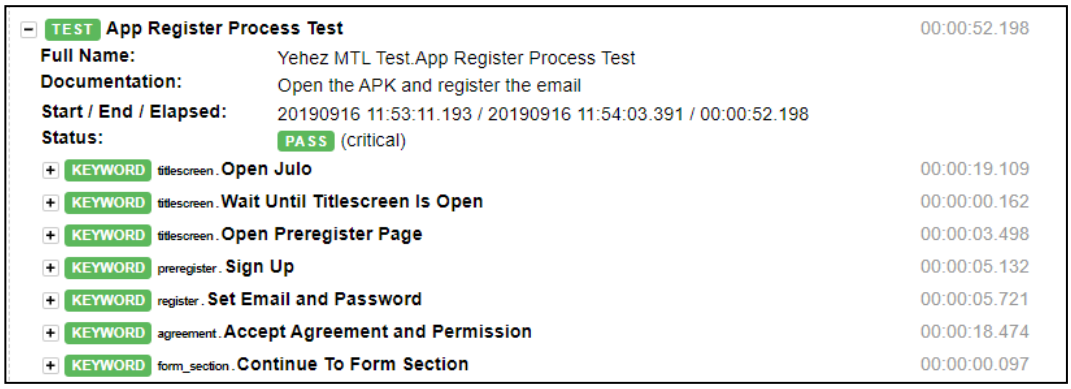
Nama Kolom	Tipe Data	Panjang	Keterangan
is_digisign_affected	bool	1	Pengecekan apakah *digital signature* terdampak
is_new_va	bool	1	Pengecekan apakah ada *virtual account* baru

3.3.4 Implementasi

Ketika proses pengujian dijalankan dengan *script*, maka *terminal* akan menampilkan status dari hasil pengujian yang dilakukan. Setelah proses pengujian selesai, hasil *test* otomatis disimpan dalam *file* log.html. Apabila *test* pada suatu halaman atau modul berhasil, maka warna indikator akan berwarna hijau, sedangkan warna indikator akan berwarna merah jika *test* pada modul tersebut gagal.

1. Hasil pengujian proses registrasi awal

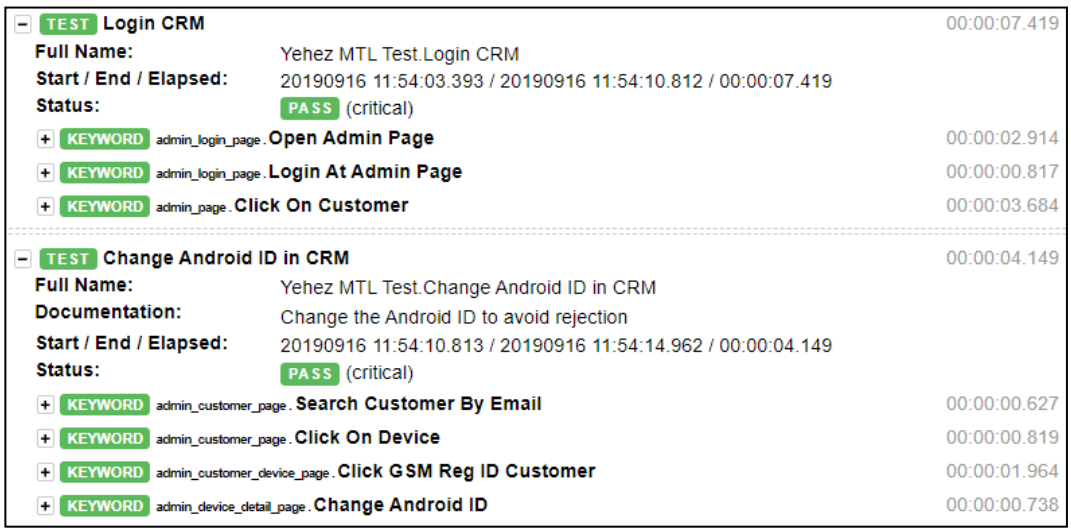
Pada Gambar 3.4 terdapat hasil pengujian yang telah dilakukan pada proses registrasi awal *user*. Selain menampilkan indicator keberhasilan *test*, di dalam log.html tersedia juga estimasi waktu yang ditempuh selama *automation test* dijalankan di dalam modul tersebut.



Gambar 3.5 Hasil pengujian proses registrasi awal

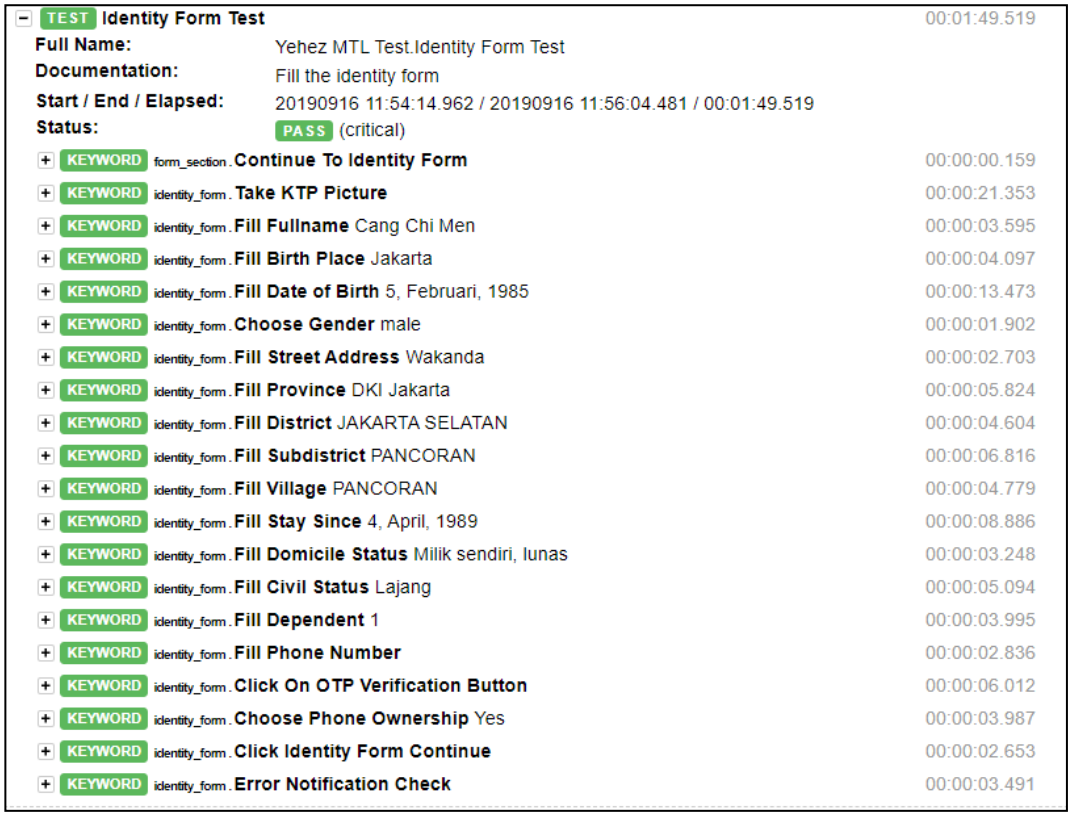
2. Hasil pengujian proses registrasi lanjutan

Setelah tahapan registrasi awal berhasil, maka akan berlanjut ke tahap registrasi lanjutan. Sebelum melanjutkan ke proses berikutnya, Android ID perlu diganti atau *dirandom* untuk keperluan *testing* dan mencegah pinjaman ditolak pada proses pengujian. Proses penggantian Android ID dapat dilihat pada Gambar 3.5.

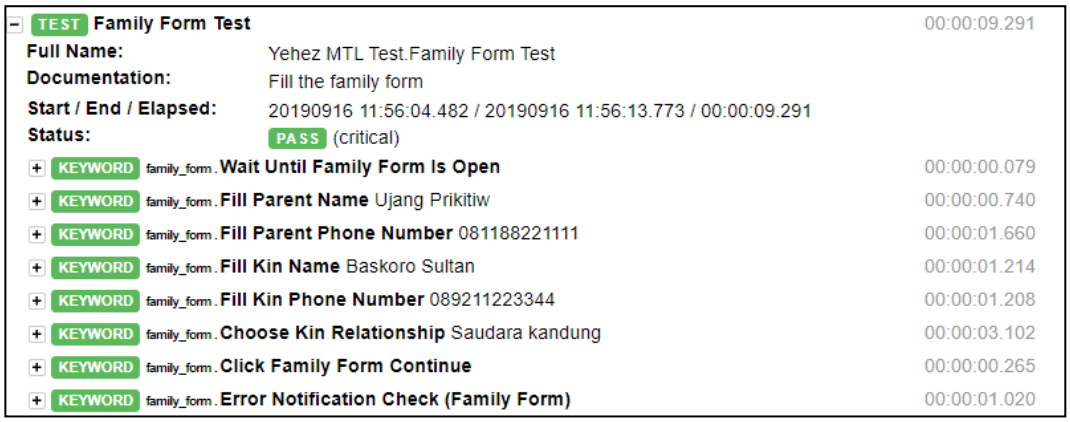


Gambar 3.6 Hasil pengujian penggantian Android ID

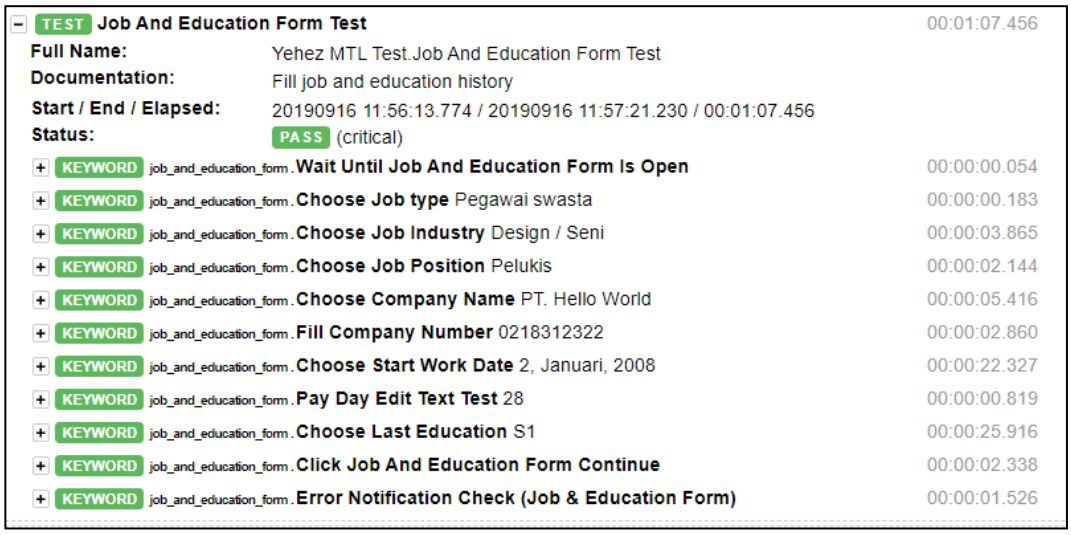
Setelah Android ID berhasil diganti, maka proses berlanjut ke tahapan registrasi lanjutan. Terdapat 4 modul utama dalam proses ini, yaitu *form* identitas *user*, *form* data keluarga, *form* pekerjaan dan pendidikan *user*, dan *form* data finansial *user*. Hasil pengujian pada modul-modul tersebut terdapat pada Gambar 3.7 sampai dengan Gambar 3.10.



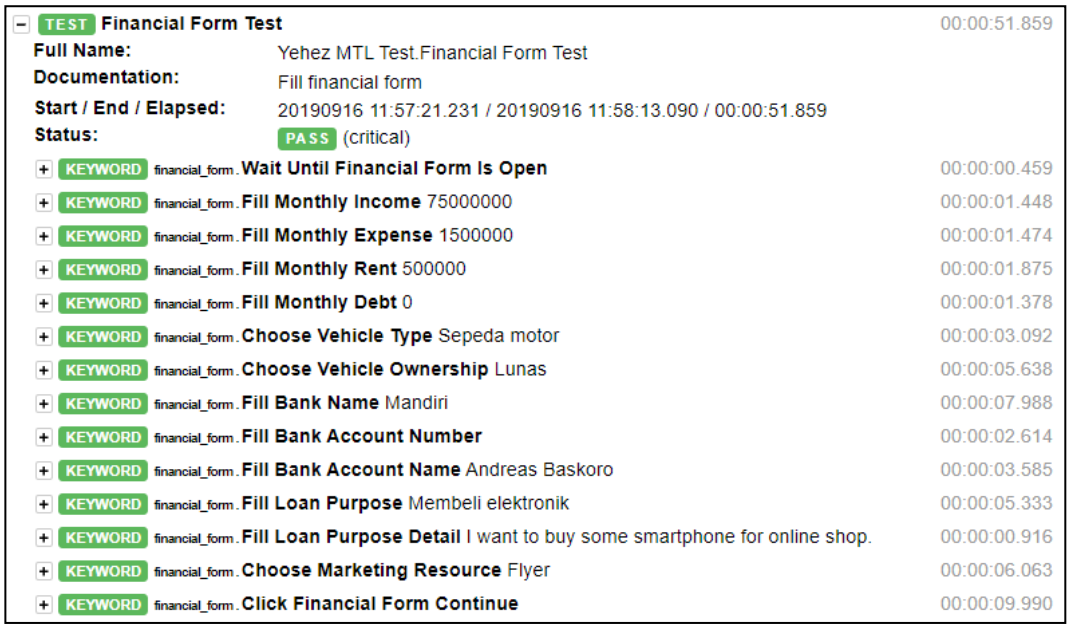
Gambar 3.7 Hasil pengujian *form* identitas *user*



Gambar 3.8 Hasil pengujian *form* data keluarga

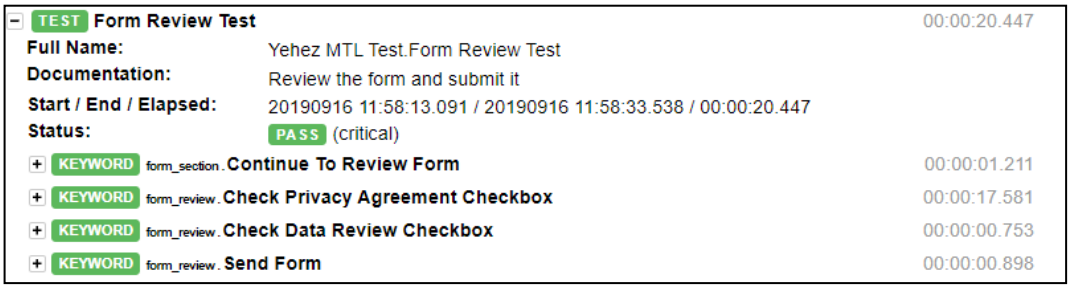


Gambar 3.9 Hasil pengujian *form* pekerjaan dan pendidikan *user*



Gambar 3.10 Hasil pengujian *form* finansial *user*

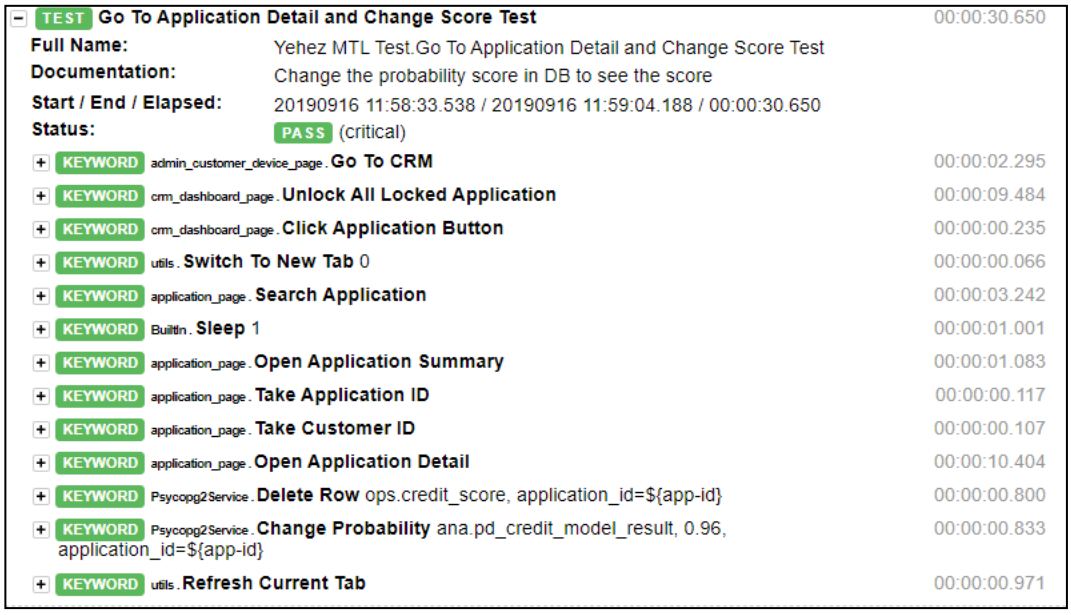
Ketika semua *form* telah terisi, maka akan dilanjutkan dengan proses *review form* untuk memastikan semua *form* telah terisi dengan benar. Pada Gambar 3.11 terdapat hasil pengujian proses *review form*.



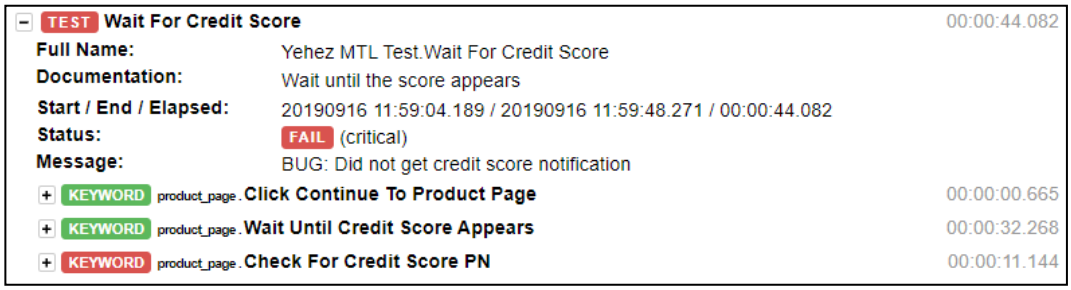
Gambar 3.11 Hasil pengujian proses *review form*

3. Hasil pengujian *credit score*

Setelah keseluruhan proses registrasi selesai, skor kredit akan didapatkan *user* berdasarkan data yang telah diberikan. Tim *data science* telah membuatkan pemodelan data yang berupa probabilitas *threshold*. Semakin tinggi probabilitas yang didapatkan *user*, maka skor yang didapat *user* akan semakin besar dan pinjaman yang diajukan pun juga semakin besar. Untuk keperluan *testing*, maka *script automation test* dimodifikasi sehingga dapat mengubah probabilitas yang telah diberikan. Pada Gambar 3.12 dan 3.13 terdapat hasil pengujian pada proses pengubahan probabilitas *threshold* dan proses menunggu skor kredit muncul di aplikasi. Pada Gambar 3.13, warna indikator berwarna merah karena terdapat *bug* yaitu tidak adanya *push notification* setelah skor kredit muncul.

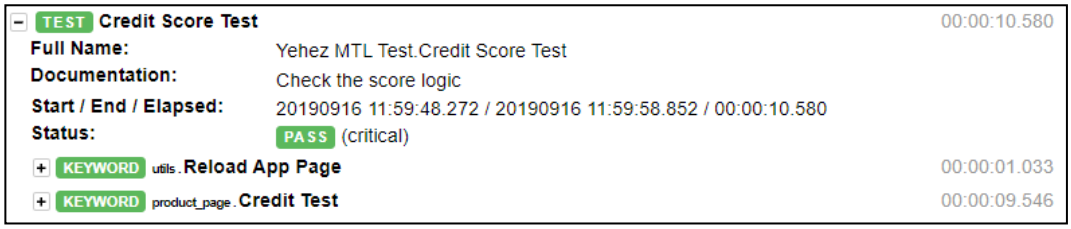


Gambar 3.12 Hasil pengujian proses mengubah probabilitas *threshold*



Gambar 3.13 Hasil pengujian proses menunggu skor kredit ditampilkan di aplikasi *mobile*

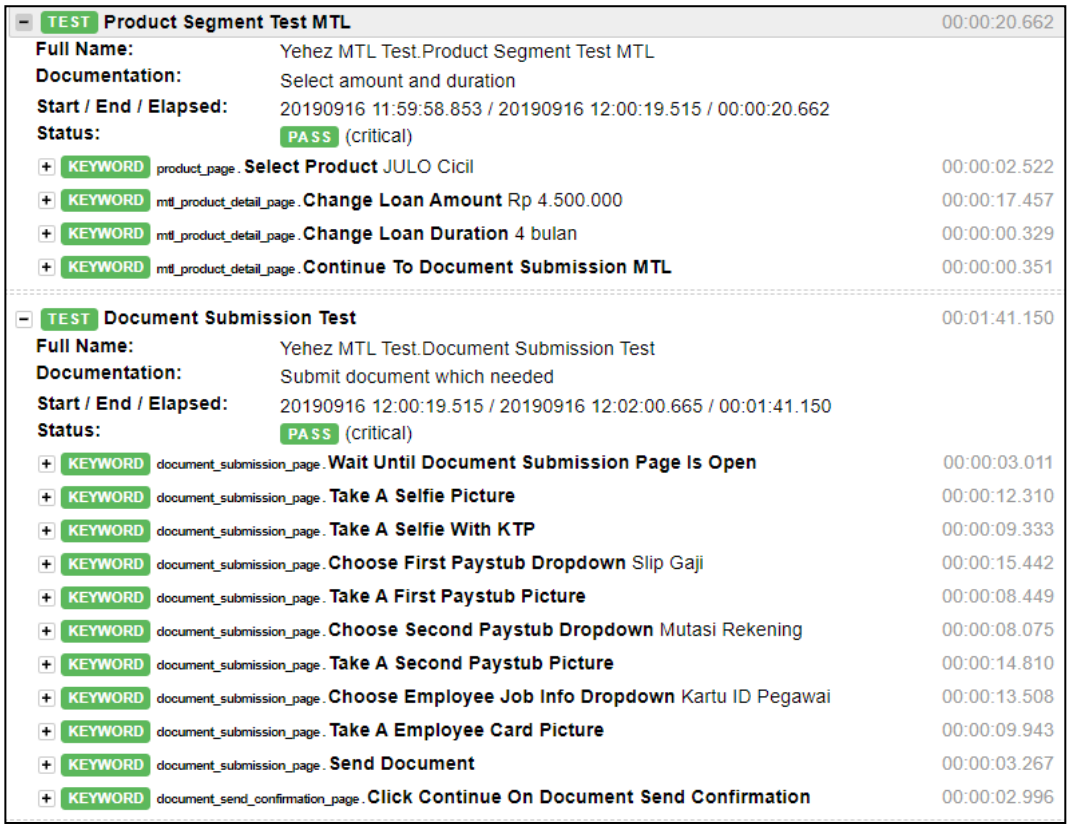
Untuk keperluan *testing*, skor yang didapatkan perlu diuji kembali. Bila skor yang didapatkan adalah C yang merupakan skor terendah, maka *user* hanya bisa mengajukan pinjaman dengan agunan (CTL). Jika skor yang didapatkan bukan C, maka *user* bisa meminjam pinjaman MTL ataupun STL namun tidak bisa mengajukan pinjaman CTL. Hasil pengujian proses tersebut terdapat pada Gambar 3.13.



Gambar 3.14 Hasil pengujian skor kredit

4. Hasil pengujian proses pemilihan produk dan *upload* dokumen

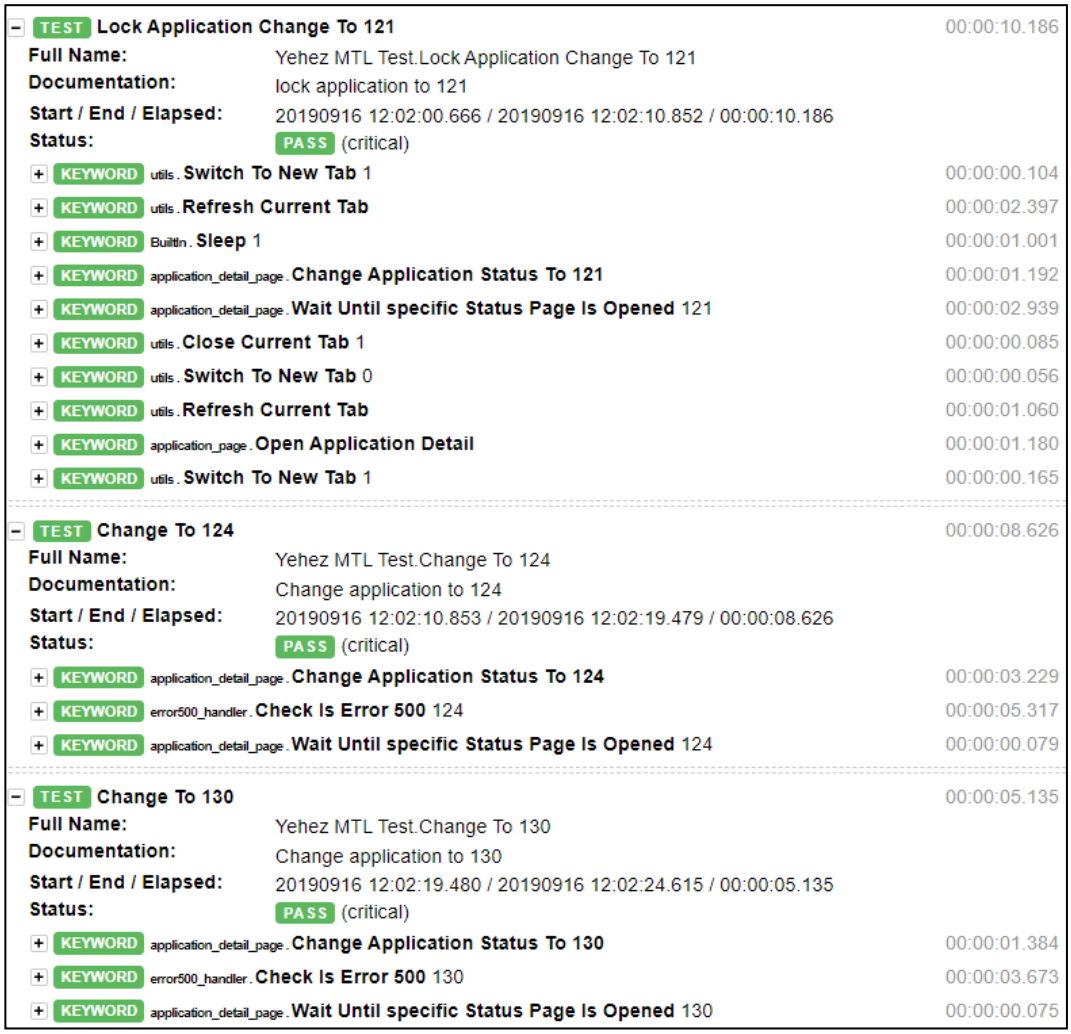
Dalam laporan ini, *user* berhasil mendapatkan skor A dan mengajukan pinjaman MTL. Dalam Gambar 3.14, *user* memilih produk JULO Cicil (MTL), memilih jumlah dan durasi pinjaman, serta *upload* dokumen yang diperlukan.



Gambar 3.15 Hasil pengujian pemilihan produk dan *upload* dokumen

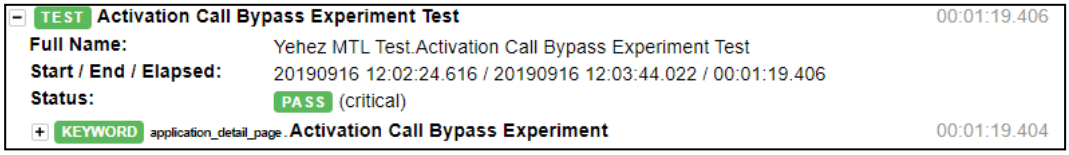
5. Hasil pengujian pemrosesan pinjaman

Setelah *user* selesai *upload* dokumen, maka *automation testing* akan berperan sebagai *agent* yang bertugas memproses pengajuan pinjaman dari *user.* Dalam proses pengujian, beberapa validasi seperti validasi KTP, *verification* call oleh agent, dan OTP nomor telepon dimatikan sementara sehingga data yang diuji diasumsikan valid. Pada Gambar 3.15 terlampir hasil pengujian perubahan status pinjaman dari 121 *(document verification)* sampai 130 *(verification success).*



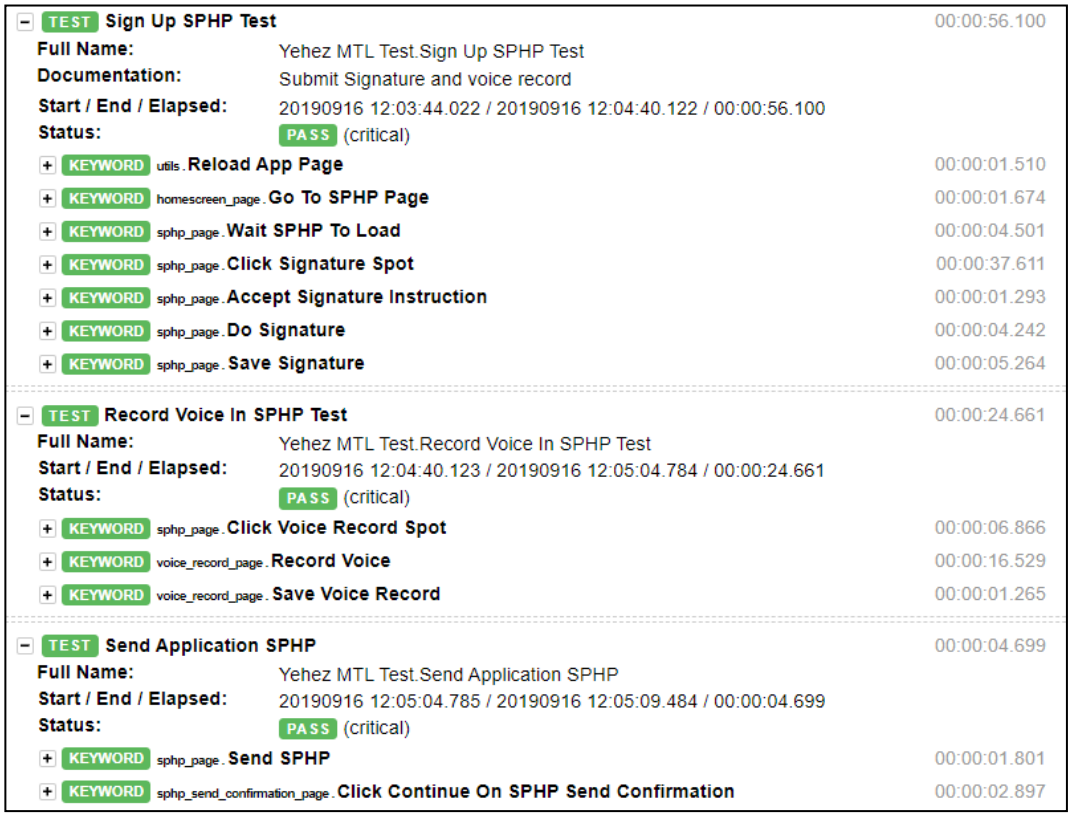
Gambar 3.16 Hasil pengujian perubahan status dari 121 sampai 130

Ketika status pinjaman sudah sampai pada status 130, maka *agent* akan melakukan *activation call* (status 141) yaitu menelepon *user* dan memberikan penawaran terkait peminjaman yang diajukan *user*. Pada proses *testing*, diasumsikan *agent* telah menyelesaikan proses tersebut (status 160). Pada Gambar 3.17 terlampir hasil pengujian perubahan status dari 130 ke 160.



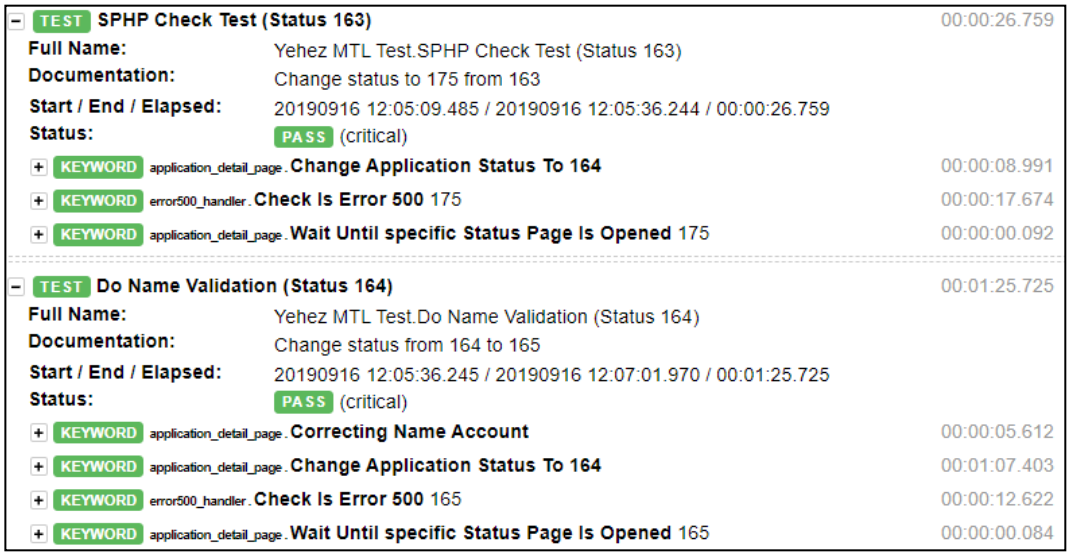
Gambar 3.17 Hasil pengujian perubahan status dari 130 menjadi 160

Setelah status pinjaman menjadi 160, maka *automation testing* kembali berperan sebagai *user*. Pada status ini, *user* harus menyetujui Surat Pemberitahuan Hasil Pemeriksaan (SPHP) yang dikirim ke aplikasi *mobile*. Selain itu, *user* juga harus mengirimkan tanda tangan *digital* dan rekaman suara sebagai tanda persetujuan. Pada Gambar 3.18 terdapat hasil pengujian dari proses tersebut.



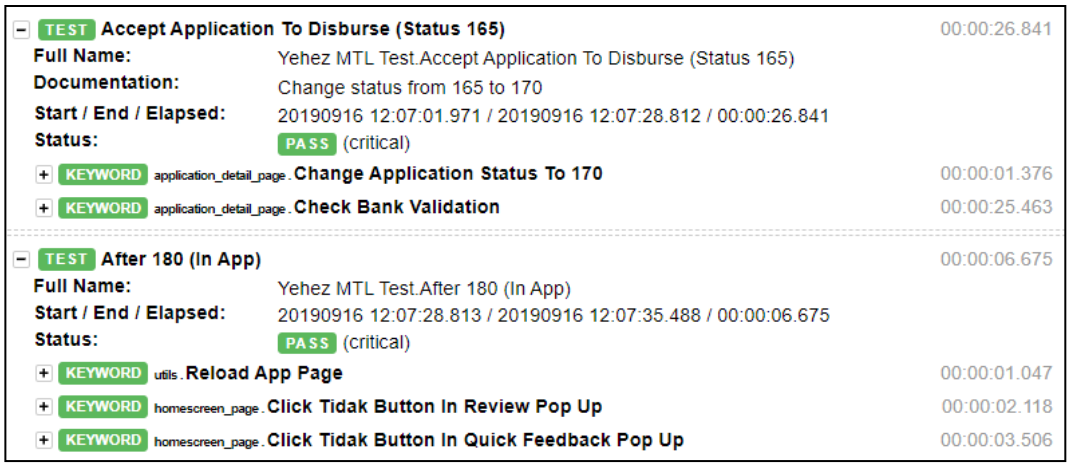
Gambar 3.18 Hasil pengujian *submit* SPHP

Setelah *user* melakukan *submit* SPHP, *agent* akan kembali melakukan verifikasi dokumen (status 163). Kemudian nama rekening bank yang tertera diganti menjadi prod_only sebagai penanda bahwa akun tersebut adalah akun *testing* (status 164). Proses tersebut terlampir pada Gambar 3.19.



Gambar 3.19 Hasil pengujian verifikasi SPHP dan penggantian nama rekening bank

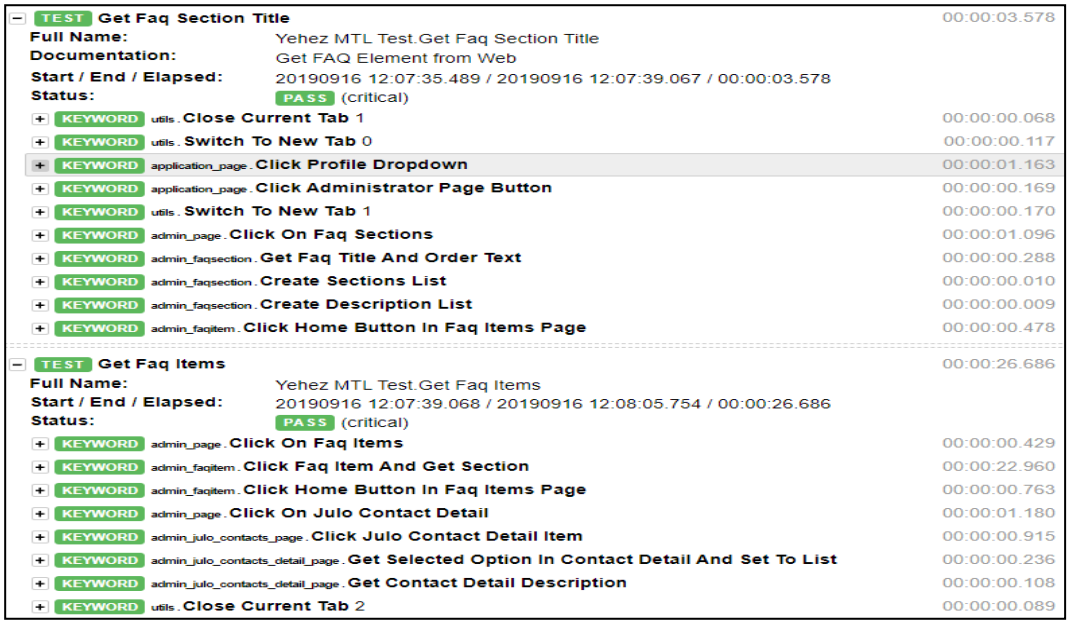
Ketika *agent* mengganti status ke 164, otomatis status akan menjadi 165 (*waiting for lender approval)* karena seluruh data dianggap valid. Begitu pula dengan proses berikutnya, *agent* diasumsikan langsung mengubah status pinjaman menjadi 170 *(legal agreement signed)* dan otomatis menjadi 180 *(fund disbursal success)*. Pada Gambar 3.20, terdapat hasil pengujian dari proses tersebut.



Gambar 3.20 Hasil pengujian pencairan dana pinjaman

6. Hasil pengujian fitur *Frequently Asked Question* (FAQ) pada web

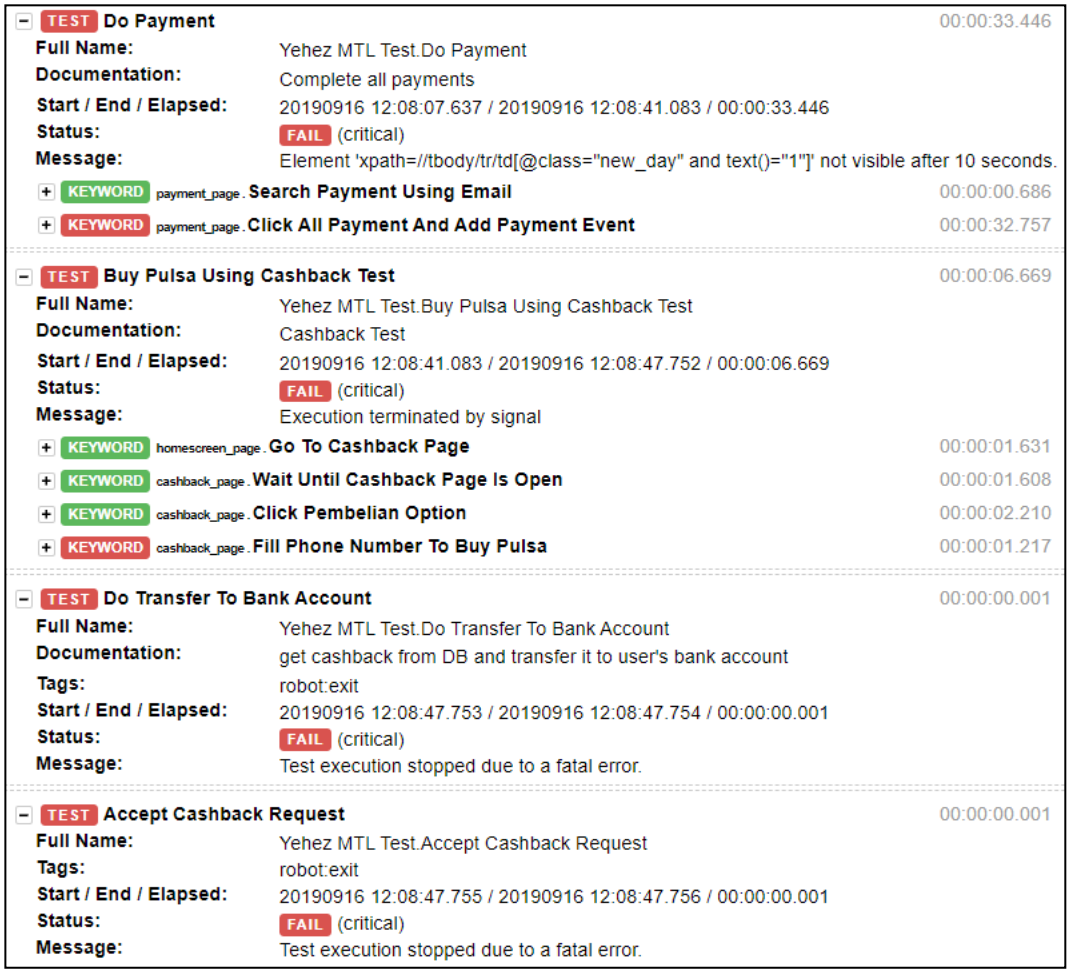
Pada proses ini, dilakukan pengecekan fitur FAQ pada *website* admin JULO. Seperti yang tertera pada Gambar 3.21, tahapan yang dilakukan cukup sederhana, yaitu *automation testing* akan membuka laman FAQ dan hanya mengecek apakah FAQ telah tersedia atau tidak.



Gambar 3.21 Hasil pengujian fitur FAQ *website* admin JULO

7. Hasil pengujian proses pembayaran dan *cashback*

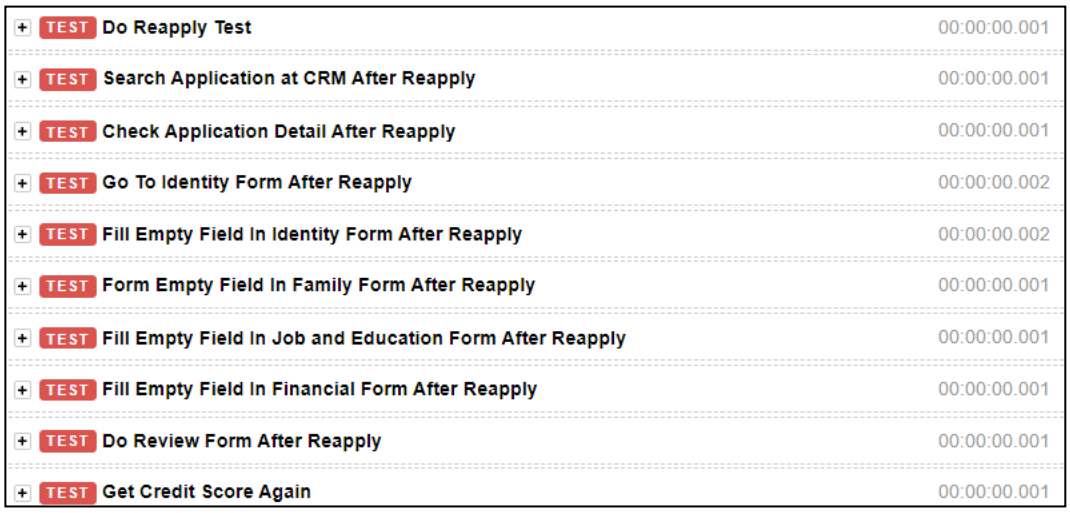
Fitur yang diuji pada proses ini sangat krusial karena menyangkut uang yang dibayarkan *user*. Pada tahap ini, *agent* diasumsikan telah menerima pembayaran dari *user*. Selain itu, jumlah yang dibayarkan sengaja dilebihkan agar *user* mendapatkan *cashback*. Pada Gambar 3.22, terdapat contoh hasil dari pengujian pembayaran sampai dengan mencairkan *cashback* ke rekening *user* yang gagal dikarenakan gangguan pada *backend server* sehingga mempengaruhi hasil test pada modul-modul berikutnya.



Gambar 3.22 Hasil pengujian proses pembayaran dan *cashback*

8. Hasil pengujian proses *reapply*

Pada proses ini diasumsikan *user* bisa melakukan pengajuan pinjaman kembali. *User* kembali melakukan registrasi ulang, hanya saja sebagian besar *form* registrasi sudah terisi dengan data-data yang sebelumnya sudah terisi saat registrasi pertama kali. Hanya ada beberapa bagian *form* yang kosong, seperti foto KTP, dan deskripsi tujuan pengajuan pinjaman sehingga proses *reapply* bisa lebih cepat dari proses *apply* pertama kali. Pada Gambar 3.23 terlampir hasil dari proses *reapply* sampai mendapatkan skor kembali dan proses ini hanya untuk memastikan proses *reapply* berjalan dengan baik tanpa dilanjutkan sampai proses pembayaran.

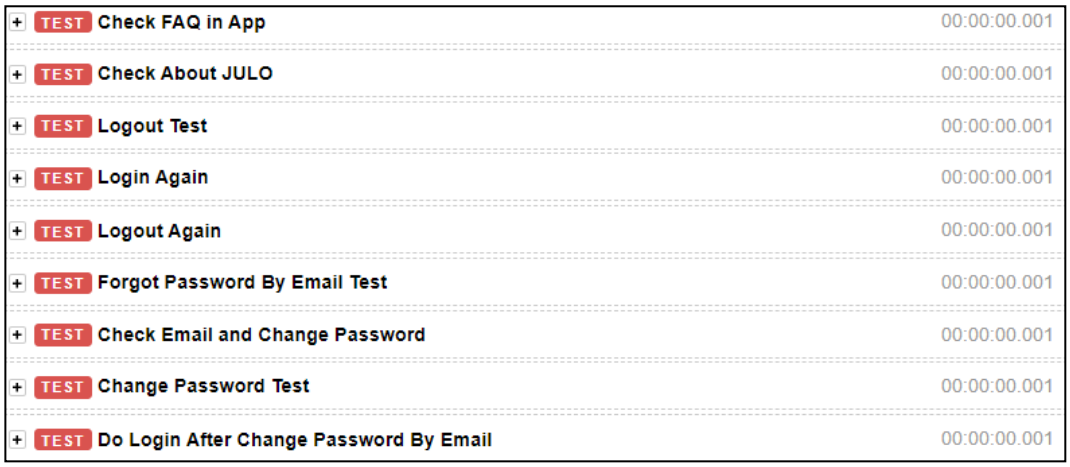


Gambar 3.23 Hasil pengujian proses *reapply*

9. Hasil pengujian proses *reset password*

Ada 2 cara untuk melakukan *reset* password pada aplikasi JULO, yaitu melalui email atau melalui *One Time Password* (OTP) yang dikirimkan lewat SMS. Pada *test* kali ini, fitur yang diuji adalah fitur *reset password* melalui email.

User mengisi *form reset password* dengan email yang digunakannya. Beberapa menit kemudian, link untuk *reset password* akan dikirimkan ke email *user*. Setelah melakukan penggantian *password* pada link yang diberikan, *user* melakukan login kembali dengan *password* baru yang telah dibuat. Pada Gambar 3.24 terlampir hasil pengujian fitur *reset password* yang gagal karena pengaruh dari kegagalan proses *test* di modul sebelumnya.



Gambar 3.24 Hasil pengujian proses *reset password*

3.3.5 Kendala yang Ditemukan

Pengembangan *automated regression test* pada aplikasi *mobile* dan *web* JULO sebenarnya tidaklah terlalu sulit. Namun, yang menjadikannya rumit adalah spesifikasi *test case* yang diberikan oleh *product manager* dan *IT Manager*. Secara umum kendala-kendala yang ditemui adalah sebagai berikut.

a. Perubahan alur kerja di tengah-tengah *development*,

b. Kurangnya komunikasi dari pihak *developer* ketika membuat perubahan elemen XML pada aplikasi *mobile* sehingga sering membuat proses pengujian gagal,

c. Banyaknya *bug* yang terdapat di sisi *backend* membuat hasil test hampir selalu tidak berhasil 100%,

d. Pembuatan *test case* masih bergantung kepada *developer* sehingga ketika belum ada *fixing* atau *release* terbaru, maka akan menghambat proses *testing*.

3.3.6 Solusi Atas Kendala yang Ditemukan

Setiap kendala yang ditemui pasti dapat dicari solusi terbaiknya. Solusi untuk menangani kendala-kendala yang telah disebutkan pada subbab sebelumnya adalah berinisiatif untuk sering berkonsultasi dengan pihak terkait. Misalnya ketika ada *release* terbaru, maka sebelum dilakukan *test* harus berkonsultasi kepada *developer* terkait.

Jika ada perubahan alur kerja di tengah *development*, bisa dikonsultasikan dengan *product manager*. Harus ada konfirmasi dan persetujuan antara *product manager* dan *developer* sehingga hasil yang didapatkan sesuai dengan spesifikasi yang dimaksudkan.