

Aplikasi Manajemen Hutang Piutang dengan OCR Berbasis Swift

Amelia Alexandra, *Teknik Informatika Institut Sains dan Teknologi Terpadu Surabaya*,
Herry Pieter, *Teknik Informatika Institut Sains dan Teknologi Terpadu Surabaya*.

Abstrak—Perkembangan aplikasi mobile yang pesat juga mendorong banyaknya aplikasi manajemen hutang perseorangan yang mulai beredar di pasaran. Saat ini, aplikasi manajemen hutang perseorangan bisa dibilang sudah cukup banyak, tetapi banyak dari aplikasi tersebut yang tidak terhubung langsung dengan perbankan. Apabila aplikasi tersebut diterapkan di era sekarang ini akan menjadi agak merepotkan, mengingat 80% orang Indonesia menyimpan sebagian besar uangnya pada rekening bank. Selain itu, akan menjadi sangat merepotkan apabila aplikasi pencatatan hutang yang digunakan tidak bisa membantu pengguna untuk mengecek mutasi masuk atau hutang mana yang sudah terbayar dan mana yang belum

Kata Kunci— aplikasi mobile ios, image processing, manajemen hutang piutang, optical character recognition,

I. PENDAHULUAN

Aplikasi pencatatan hutang piutang membuat proses pencatatan hutang akan menjadi lebih rapi dan teratur dikarenakan setiap hutang dapat dicatat dengan baik dan bahkan diverifikasi oleh kedua belah pihak (pihak yang memberi hutang dan pihak yang berhutang), serta setiap pembayaran hutang dapat didokumentasikan dengan baik. Berdasarkan perkembangan teknologi dan masalah di atas, akan dibuat sebuah aplikasi manajemen hutang untuk mencatat segala informasi mengenai hutang dan piutang orang tersebut, mulai dari hutang yang diberikan kepada orang lain, pembayaran yang telah dilakukan oleh pihak yang berhutang, sampai rekapitulasi serta laporan mengenai perhitungan hutang secara menyeluruh dalam suatu periode tertentu.

II. TEORI DASAR

Aplikasi Manajemen Hutang Piutang dengan OCR Berbasis iOS, seperti swift dengan Xcode, OCR Library, Charts Library, Web Service, API Perbankan, dan Firebase. Berikut ini penjelasan framework dalam bentuk poin-poin.

A. Swift

Swift adalah bahasa pemrograman yang kokoh dan intuitif yang diciptakan Apple untuk mengembangkan aplikasi untuk iOS, Mac, Apple TV, dan Apple Watch. Bahasa ini didesain untuk memberikan lebih banyak

kebebasan kepada pengembang. Swift mudah digunakan dan bersifat sumber terbuka (open source).

Swift adalah hasil penelitian terbaru tentang bahasa pemrograman, dikombinasikan dengan pengalaman puluhan tahun membangun platform Apple. Parameter diekspresikan dalam sintaks bersih yang membuat API di Swift lebih mudah dibaca. Lebih baik lagi, tidak perlu mengetikkan titik koma setiap akhir sintaks.

Pada pembuatan aplikasi Ios menggunakan swift, pengguna disarankan untuk menggunakan IDE bawaan yaitu *xcode*. Pada *xcode* pengguna akan diberikan peringatan jika variabel yang dibuat tidak efisien, contohnya pada saat pembuatan variabel menggunakan *var* dan *let*. Variabel yang menggunakan *var* memakan memory yang lebih besar dibanding *let*, karena isi dari *var* dapat diubah oleh pengguna. Sedangkan isi dari variabel *let* sudah tidak dapat diubah atau hanya sekali pengisian.

Pada Swift language pengguna juga dapat menggunakan library yang berbasis pada bahasa C, sehingga library yang dapat digunakan sangat banyak. Untuk menggunakan library berbasis C, pengguna perlu membuat file header sebagai jembatan agar bisa digunakan oleh swift. File header tersebut bertujuan untuk mengubah library menjadi object yang bisa digunakan[4].

Salah satu kelemahan dari swift language ialah pada saat terdapat versi baru, function yang digunakan di versi lama, sebagian akan menjadi deprecated dan menyebabkan kebanyakan kode yang telah dibuat menjadi error dan perlu diupdate, sehingga pembuat aplikasi harus mempelajari function baru setiap adanya versi baru pada swift.

B. OCR

Pada subbab ini akan dijelaskan secara singkat mengenai OCR. OCR merupakan singkatan dari *Optical Character Recognition* merupakan teknik yang digunakan untuk mengubah tulisan pada gambar menjadi sebuah teks yang dapat dikenali atau dibaca oleh aplikasi. Pada pembuatan OCR terdapat dua algoritma yang sering digunakan, berikut ini algoritma tersebut[1].

1. Pattern Recognition

Algoritma ini membutuhkan teks atau tulisan yang digunakan sebagai pembanding, sehingga pada saat proses pengenalan karakter dapat membedakan mana yang merupakan teks dan mana yang bukan merupakan teks.

2. Feature Detection

Algoritma ini digunakan untuk mendeteksi sebuah karakter yang lebih spesifik. Misalnya sebuah garis lurus, garis melengkung ataupun mengerucut. Sehingga hasil yang didapatkan akan lebih bagus dan optimal.

Herry Pieter, Departemen Teknik Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: herrypieter30@gmail.com)

Amelia Alexandra, Departemen Teknik Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: amelia@stts.edu)

Saat ini OCR telah banyak digunakan dalam aplikasi sehari-hari seperti aplikasi editing, scanner gambar, aplikasi lalu lintas dll. OCR mempunyai beberapa tahapan proses yang sering digunakan atau dijadikan dasar dalam pembuatan aplikasi OCR, berikut tahapan kerja tersebut.

1. Meratakan Gambar
Pada proses ini gambar yang miring akan dibuat sejajar atau lurus.
2. Melakukan Analisa
Bagian ini merupakan proses untuk mengidentifikasi mana yang gambar dan mana yang teks.
3. Mengatur Arah Gambar
Pada bagian ini aplikasi akan mengambil sebagian area pada gambar untuk mengidentifikasi sudut gambar.
4. Memisahkan Karakter Huruf dan Angka
Pada bagian ini proses yang dilakukan adalah memisahkan mana yang merupakan huruf dan angka.
5. Mengidentifikasi Gambar
Pada bagian ini merupakan proses mengidentifikasi satu persatu gambar yang sudah dipisahkan dan mengubah menjadi teks.
6. Membuat Hasil Akhir
Setelah angka dan huruf dipisahkan, proses yang terakhir dilakukan ialah menyatukan kembali bagian yang merupakan teks dan menjadikan file yang diinginkan.

C. Charts

Pada subbab ini akan dijelaskan mengenai charts dari bahasa pemrograman swift yang akan digunakan. Charts ini digunakan untuk menampilkan sekumpulan data yang ingin dikonversikan menjadi grafik-grafik sehingga memudahkan data tersebut untuk dibaca dan dipahami. Berikut ini adalah fitur-fitur yang disediakan oleh Charts ini.

- 1) Penskalaan pada kedua sumbu yang dapat dilakukan dengan touch-gesture, atau pinch zoom
- 2) Charts dapat digeser-geser
- 3) Charts dapat dikombinasikan contoh (line-, bar-, scatter-, candle-stick-, bubble-)
- 4) Sumbu dapat dikustom sesuai dengan kebutuhan
- 5) Charts dapat disimpan menjadi gambar yang dapat berupa PNG atau JPG
- 6) Charts ini dapat memiliki animasi sesuai keinginan
- 7) Sangat dapat dikustom dari warna, jenis font, legenda, warna, latar belakang, gesture, dan lain sebagainya

Charts mempunyai kelebihan dalam proses implementasi yang disebabkan karena charts merupakan library bawaan dari swift, sehingga pembuat tinggal memanggil library tersebut untuk menggunakan charts. Charts juga mempunyai delapan macam bentuk grafik, dimana grafik tersebut pada umumnya dibutuhkan dalam laporan. Delapan grafik tersebut yaitu :

1. Line Chart (with legend, simple design)
2. Bar Chart (simple design)
3. Combined Chart (bar and line)
4. Pie Chart
5. Scatter Chart
6. Candle Stick Chart
7. Buble Chart
8. Radar Chart

Pada masing-masing jenis chart juga terdapat pilihan yang bisa dicustom, seperti pada bar chart, dimana pengguna bisa membuat simple, horizontal dan dataset bar chart. Pada aplikasi manajemen hutang piutang, jenis grafik yang dipakai ialah pie chart. Cara penggunaan pie chart cukup dengan memasukkan array yang berisi label beserta nominal yang didapatkan ke dalam dataset charts. Kemudian setelah mengatur dataset, pengguna perlu melakukan reload ke object chart agar charts menampilkan data yang didapatkan.

D. Web Service

Pada subbab ini akan dijelaskan mengenai web service. Web service pada aplikasi manajemen hutang piutang ini adalah sebagai perantara antara database dengan aplikasi manajemen hutang piutang ini. Web service yang digunakan berbasis REST API. Berikut adalah penjelasan mengenai web service yang digunakan pada tugas akhir ini.

REST (Representational State Transfer) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan terutama dalam pertukaran dan komunikasi data[5].

Pada subbab ini akan dijelaskan mengenai web service. Web service pada aplikasi manajemen hutang piutang ini adalah sebagai perantara antara database dengan aplikasi manajemen hutang piutang ini. Web service yang digunakan berbasis REST API. Berikut adalah penjelasan mengenai web service yang digunakan pada tugas akhir ini.

REST (Representational State Transfer) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan terutama dalam pertukaran dan komunikasi data.

Slim framework sangat mudah digunakan oleh pengguna, karena memiliki kerangka yang telah disediakan. Sehingga pengguna hanya perlu mengatur username, password, dan database yang akan digunakan. Slim framework sangat cocok digunakan untuk proyek skala kecil hingga menengah, karena proses pemasangan yang tidak lebih dari 5 menit[6].

Kelemahan dari slim framework pada saat pengguna ingin membuat object oriented. Proses pengaturan yang diperlukan lebih rumit dibanding dengan pembuatan menggunakan laravel atau lumen. Slim framework menggunakan query SQL dan tidak menyediakan query builder, seperti web service framework lainnya.

E. Face ID

Pada subbab ini akan dijelaskan mengenai Face ID yang merupakan fitur keamanan buatan Apple. Face ID dirilis pada November 2017 bersamaan dengan dirilisnya iPhone X. Fitur ini dikembangkan untuk menjadi pengganti Touch ID. Face ID bekerja dengan menggunakan camera *true depth* yang memindai 30.000 titik infra merah yang digunakan untuk memetakan wajah pengguna menjadi gambar 2D.

Gambar ini disimpan di dalam *secure enclave* secara lokal dalam ponsel tiap pengguna. Face ID menggunakan

beberapa sistem keamanan yang bertujuan untuk mencegah *spoofing*, sehingga Face ID dapat mencegah pengguna lain yang ingin membuka ponsel dengan topeng dan foto [8].

Face ID dapat mengenali pengguna walaupun menggunakan aksesoris tambahan seperti topi, kacamata, bahkan kumis dan jenggot. Kelemahan utama Face ID ialah tidak dapat membedakan pengguna apabila memiliki kembaran yang mirip. Face ID diklaim hanya dapat dibobol oleh satu orang berbanding satu juta karena menyimpan hingga 300.000 titik muka yang sangat unik.

Pada aplikasi manajemen hutang piutang, Face ID digunakan sebagai salah satu security untuk dapat membuka aplikasi. Untuk dapat menggunakan Face ID dalam aplikasi manajemen hutang piutang, pengguna harus mengizinkan aplikasi untuk mengakses Face ID pada perangkat dan juga pengguna harus mengaktifkan Face ID di perangkat tersebut.

Pada Swift Language, implementasi Face ID dan Touch ID menggunakan module yang sama. Namun, untuk dapat menggunakan Touch ID, pengembang perlu mengatur hak akses yang berbeda dengan Face ID, dan juga handler yang digunakan untuk mengolah data yang dikembalikan Touch ID berbeda dengan Face ID.

F. Api Perbankan

Pada subbab ini akan dijelaskan mengenai API Perbankan. API Perbankan adalah API yang digunakan untuk mengecek status transaksi dari pengguna. Untuk aplikasi manajemen hutang piutang ini, API Perbankan oada aplikasi manajemen hutang piutang ini digunakan untuk validasi pembayaran utang atau piutang via transfer.

Cekmutasi.co.id merupakan sebuah web-based software terintegrasi untuk mengelola banyak rekening dalam satu dashboard. Sistem ini dilengkapi robot mutasi yang bekerja memeriksa transaksi terbaru dalam rekening pengguna dan menyajikannya dalam sebuah laporan sehingga pengguna tidak perlu melakukan login internet banking berulang-ulang hanya untuk mengecek pembayaran.

Sistem cekmutasi juga mendukung proses validasi pembayaran otomatis yang dapat diintegrasikan ke bisnis pengguna melalui Restful API. Cekmutasi sendiri sudah support dengan Bank BRI, BNI, BCA, Mandiri, PayPal, OVO & GO-PAY [10]. Cekmutasi.co.id juga sudah menyediakan url callback untuk memudahkan proses integrasi, berikut contoh penggunaan API Perbankan pada aplikasi manajemen hutang piutang ini.

Pada aplikasi manajemen hutang piutang, fitur cekmutasi.co.id yang digunakan ialah notifikasi callback. Pembuat aplikasi dapat memasukkan url API yang telah dibuat pada saat pendaftaran rekening di dashboard cekmutasi.co.id, kemudian mengatur interval bot cekmutasi.co.id untuk melakukan crawl ke internet banking masing-masing bank. Jika bot pada mutasi.co.id menemukan mutasi baru pada internet banking, cekmutasi.co.id akan mengirim notifikasi ke url callback yang telah diatur. Data JSON yang dikirimkan oleh cekmutasi.co.id akan diproses oleh web service untuk melakukan pengecekan request pembayaran oleh pengguna.

Kelemahan dari penggunaan cekmutasi.co.id ialah proses yang dilakukan tidak real time, karena terdapat delay pada interval crawl yang dilakukan. Jika interval crawl terlalu kecil, resiko akun internet banking pengguna akan diblock

lebih besar. Interval yang disarankan untuk melakukan crawl ialah setiap 3-5 menit sekali.

G. Firebase Notification

Menggunakan Firebase Cloud Messaging (FCM) dapat memberi tahu aplikasi klien bahwa email baru atau data lain tersedia untuk disinkronkan. Pengguna dapat mengirim pesan notifikasi untuk mendorong keterlibatan dan retensi pengguna[3].

Untuk kasus-kasus penggunaan seperti pengiriman pesan instan, sebuah pesan dapat mentransfer muatan hingga 4KB ke aplikasi klien. Implementasi FCM mencakup dua komponen utama untuk mengirim dan menerima pesan. Dua komponen utama tersebut ialah.

- 1) Environment yang terpercaya seperti firebase cloud, atau sebuah server aplikasi yang digunakan untuk membuat, dan mengirim pesan.
- 2) Aplikasi klien berbasis iOS, Android, atau web (JavaScript) yang menerima pesan melalui layanan platform terkait.

Firebase memiliki kelebihan yaitu proses implementasi yang mudah dan dokumentasi yang lengkap. Firebase juga memiliki kecepatan dalam proses pengirim notifikasi yang sangat cepat dan menggunakan data yang lebih kecil dari yang lain. Untuk melakukan pemasangan notifikasi firebase ke aplikasi iOS pengguna memerlukan akun developer dari Apple. Akun tersebut digunakan untuk mengeneratekan certificate untuk push notification.

Certificate tersebut akan dimasukkan ke dashboard firebase agar pengguna dapat mengirim push notification. Firebase juga menyediakan fitur scheduled notification, dimana pembuat aplikasi dapat mengatur tanggal serta jam yang diinginkan[2].

H. Preprocessing Gambar

Pada subbab ini akan dijelaskan mengenai preprocessing gambar. Preprocessing merupakan suatu bentuk pengolahan atau pemrosesan dengan input gambar dan diubah menjadi suatu gambar lain dengan menggunakan teknik tertentu. Image preprocessing dilakukan untuk memperbaiki serta meningkatkan kualitas gambar agar lebih mudah dilihat oleh manusia dan proses manipulasi terhadap gambar [9]. Image preprocessing terbagi menjadi beberapa kelompok berdasarkan dari tujuan transformasi yang diinginkan. Berikut ini kelompok dari Image Processing.

1. Image Enhancement
Proses ini berfungsi untuk meningkatkan fitur tertentu pada citra, sehingga tingkat keberhasilan dalam pengolahan gambar menjadi lebih tinggi. Contoh metode image enhancement seperti contrast stretching, gray scalling, linear contrast adjusment dan histogram equalization.
2. Image Restoration
Proses ini bertujuan untuk mengembalikan kondisi gambar yang sudah rusak atau cacat karena adanya gangguan yang menyebabkan penurunan kualitas gambar. Contoh metode yang bisa digunakan dalam kelompok ini adalah median filter, adaptive filter, linear filter dan block matching.
3. Image Compression
Proses ini bertujuan untuk membuat ukuran gambar menjadi lebih kecil dengan memadatkan bit pada citra. Proses ini dilakukan agar pengiriman gambar

menjadi lebih cepat. Contoh teknik yang umum digunakan untuk image compression ialah. Fractal, wavelets, chroma sub-sampling dan transform coding.

4. Image Representation & Modelling

Proses ini merupakan pengelompokan batas-batas gambar yang ingin digunakan. Metode yang sering digunakan untuk image representation & modelling ialah block pattern.



Gambar. 1. Contoh Sesudah dan Sebelum Contrast Stretching

Pada gambar 1 merupakan perbedaan gambar asli dan setelah dilakukan proses contrast stretching, gambar yang menggunakan contrast stretching membuat content di dalam gambar lebih jelas dan mudah dikenali. Oleh karena itu pre processing gambar sangat dibutuhkan didalam aplikasi ini.

Pada aplikasi manajemen hutang piutang, kelompok image preprocessing yang digunakan ialah image enhancement. Metode ini digunakan karena bertujuan untuk meningkatkan kualitas gambar nota sebelum dikirim ke Google Vision dan memberi hasil yang lebih optimal. Metode image enhancement yang digunakan ialah gray scalling dan contrast stretchng.

III. ANALISA SISTEM

Pada tahap ini dijelaskan tentang analisa sistem dari aplikasi manajemen hutang piutang. Penjelasan berupa deskripsi sistem yang dibuat, deskripsi dari sistem pada aplikasi-aplikasi referensi analisa permasalahan yang ada, dan spesifikasi kebutuhan dari sistem aplikasi yang dibuat.

Aplikasi Splitwise memili beberapa fitur, seperti menambah utang, melihat total utang dan piutang keseluruhan, selain itu terdapat fitur untuk menyederhanakan utang piutang di dalam group. Pada fitur dari Splitwise user dapat melakukan register melalui email dan login menggunakan google, setelah user melakukan login, user dapat melihat total hutang dan piutang pada halaman dashboard aplikasi tersebut.

Untuk membayar hutang ke pengguna lain, pengguna dapat masuk ke halaman teman atau grup dan terdapat tombol "Settle up" yang berfungsi melunaskan hutang ke pengguna lain seperti pada gambar 3.4, setelah itu pengguna satu nya akan mendapat notifikasi bahwa hutang sudah dibayar oleh pengguna lain. Jika pengguna tersebut merasa tidak menerima pembayaran, maka si pengguna dapat menghapus pembayaran yang dibuat oleh pengguna lain.

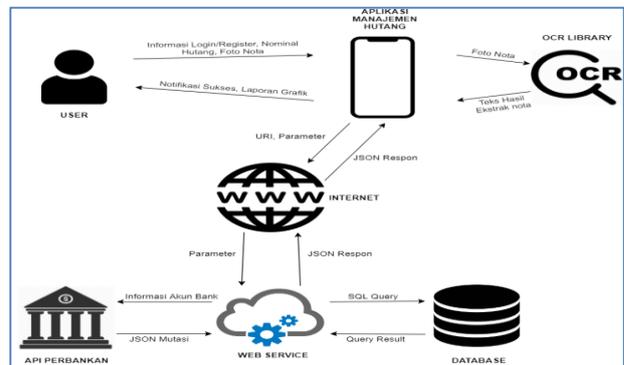
akan dijelaskan hasil spesifikasi kebutuhan yang diperlukan sistem aplikasi manajemen utang piutang. Spesifikasi kebutuhan ini berupa fitur-fitur yang menjadi solusi dari aplikasi referensi. Fitur-fitur yang menjadi solusi ini menjadi fitur utama pada aplikasi manajemen hutang piutang. Berikut dijelaskan fitur-fitur yang menjadi solusi dan fitur pendukung lain tersebut dalam bentuk poin-poin.

- Input Cicilan

- Pelunasan Hutang Otomatis
- Konfirmasi Hutang
- Kategori dan Keterangan Hutang
- Laporan Hutang
- Verifikasi Pembayaran Hutang Otomatis
- Chat
- Scan Nota
- Scan Bukti Pembayaran
- Promo
- Security Password

IV. DESAIN

Pada tahap ini akan dijelaskan desain-desain yang diterapkan pada program aplikasi manajemen hutang piutang untuk aplikasi manajemen hutang piutang ini. Penjelasan berupa desain arsitektur, desain database, desain tampilan.



Gambar. 2. Arsitektur Sistem

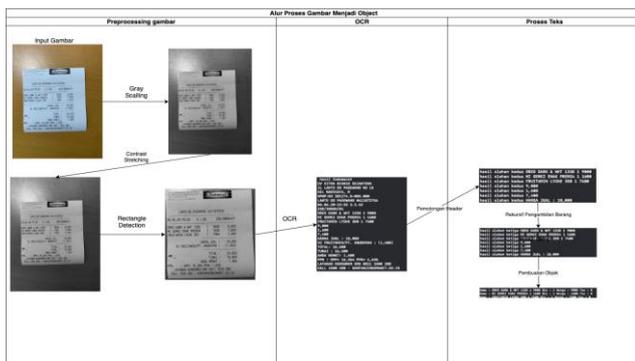
Seperti dapat dilihat pada Gambar 2, alur dari aplikasi ini dimulai ketika user membuka aplikasi. Saat aplikasi dibuka pertama kali, maka akan diberikan tawaran apakah user sudah memiliki akun atau belum. Apabila user belum memiliki akun, maka user dapat melakukan register terlebih dahulu. Namun, ketika user telah memiliki akun, maka user dapat langsung masuk dengan akun yang telah dimiliki. Ketika user telah berada dalam halaman home, maka user sudah dapat menggunakan fitur dari aplikasi ini dengan utuh. User mulai dapat menambahkan hutang dengan cara klik menu tambah hutang, pilih teman yang hendak ditunjuk sebagai pihak yang berhutang, dan nominal. Selain dengan menginputkan manual, user juga dapat melampirkan foto dari nota belanjanya. Dengan melampirkan foto nota belanja tersebut, maka aplikasi akan secara otomatis memprosesnya sebagai hutang baru. Foto yang dilampirkan di dalam aplikasi tersebut akan diproses oleh OCR Library untuk diekstrak menjadi informasi yang lebih siap diolah dalam aplikasi[7].

Setiap data hutang tersebut akan disinkronisasikan secara berkala dengan bantuan internet. Internet akan berguna untuk menghubungkan aplikasi dengan server. Koneksi dengan server sangat diperlukan untuk melihat dan menyimpan daftar hutang yang tersimpan di dalam server. Selain untuk mengelola daftar hutang, koneksi dengan internet juga diperlukan apabila aplikasi membutuhkan koneksi dengan akun bank yang dibantu oleh web service dan API Perbankan yang memudahkan pengguna.

Seperti dapat dilihat pada Gambar 1, alur dari aplikasi ini dimulai ketika user membuka aplikasi. Saat aplikasi dibuka pertama kali, maka akan diberikan tawaran apakah user

sudah memiliki akun atau belum. Apabila user belum memiliki akun, maka user dapat melakukan register terlebih dahulu. Namun, ketika user telah memiliki akun, maka user dapat langsung masuk dengan akun yang telah dimiliki. Ketika user telah berada dalam halaman home, maka user sudah dapat menggunakan fitur dari aplikasi ini dengan utuh. User mulai dapat menambahkan hutang dengan cara klik menu tambah hutang, pilih teman yang hendak ditunjuk sebagai pihak yang berhutang, dan nominal. Selain dengan menginputkan manual, user juga dapat melampirkan foto dari nota belanjanya. Dengan melampirkan foto nota belanja tersebut, maka aplikasi akan secara otomatis memprosesnya sebagai hutang baru. Foto yang dilampirkan di dalam aplikasi tersebut akan diproses oleh OCR Library untuk diekstrak menjadi informasi yang lebih siap diolah dalam aplikasi.

Setiap data hutang tersebut akan disinkronisasikan secara berkala dengan bantuan internet. Internet akan berguna untuk menghubungkan aplikasi dengan server. Koneksi dengan server sangat diperlukan untuk melihat dan menyimpan daftar hutang yang tersimpan di dalam server. Selain untuk mengelola daftar hutang, koneksi dengan internet juga diperlukan apabila aplikasi membutuhkan koneksi dengan akun bank yang dibantu oleh web service dan API Perbankan yang memudahkan pengguna sehingga tidak diperlukan untuk mengecek sendiri.



Gambar. 5. Flow Gambar Menjadi Object

Pada subbab ini akan dijelaskan mengenai alur input gambar menjadi objek yang akan digunakan di dalam aplikasi manajemen hutang piutang. Pada gambar 5 terdapat tiga bagian proses, yaitu preprocessing image, OCR yang dilakukan oleh Google Vision, dan proses yang dilakukan terhadap teks hasil OCR.

Pada proses preprocessing, langkah pertama yang dilakukan adalah gambar akan diubah dengan gray scalling, kemudian akan dilakukan contrast stretching agar gambar yang blur menjadi lebih jelas, serta garis pada nota menjadi lebih terlihat. Proses terakhir yang dilakukan pada preprocessing ialah rectangle detection untuk membuang background pada gambar dan berfokus ke nota. Pada rectangle detection, batasan gambar nota yang bisa diproses apabila keempat sudut nota tidak sampai ujung gambar.

Pada proses selanjutnya, gambar yang telah diproses akan dikirim ke Google Vision untuk melakukan OCR. Google Vision akan melakukan proses layouting dan pengenalan karakter pada gambar yang dikirim, kemudian Google Vision akan mengembalikan hasil berupa string yang berhasil dibaca. String tersebut akan diproses untuk membuang baris yang tidak terpakai.

Proses pengolahan teks yang pertama ialah pemotongan header. Proses ini bertujuan untuk membuang semua kalimat yang ada pada header nota, sehingga mempermudah dalam pengambilan nama barang. Dalam proses pemotongan header terdapat beberapa step yang dilakukan, step tersebut sebagai berikut.

1. Pencarian index kalimat pertama yang mengandung harga dan index lebih besar dari 3 dalam teks yang dihasilkan oleh Google Vision dengan menggunakan regex `'[0-9][.][0-9]'` dan regex `'@[0-9][.][0-9]'`.
2. Index kalimat pertama yang mengandung harga menjadi batas akhir dalam proses looping mencari index terbesar dari kalimat yang mengandung pola jam, tanggal, kode nota dan salah huruf akibat kesalahan pengenalan karakter oleh Google Vision. Regex yang digunakan pencarian format header yang sudah ditentukan ialah `'[0-9][:]/[0-9]'`, `'[*?!=]/'` dan `'^\\s*([a-zA-z]{1})$'`
3. Index yang didapatkan dari proses looping point nomor 2 akan ditambahkan satu, kemudian hasil tersebut akan digunakan untuk menjadi batasan akhir dalam proses penghapusan isi array.

Kelemahan utama dalam proses pemotongan header, yaitu jika pada header nota terdapat pola yang sesuai dengan regex yang sudah ditentukan. Seperti terdapat nomor npwp pada header atau kode nota yang mempunyai format seperti harga, yang menyebabkan index batasan yang didapat bukan index harga pertama dalam suatu nota.

Setelah proses pemotongan header selesai, maka akan dilanjutkan ke pemotongan footer, yang bertujuan untuk membuang kalimat yang ada pada footer, sehingga data yang terdapat pada array hanya data barang beserta atributnya. Dalam proses pemotongan footer akan dilakukan looping untuk mencari kalimat pertama atau index terkecil dalam array yang mengandung pola atau kata yang telah ditentukan, pola atau kata yang digunakan sebagai berikut.

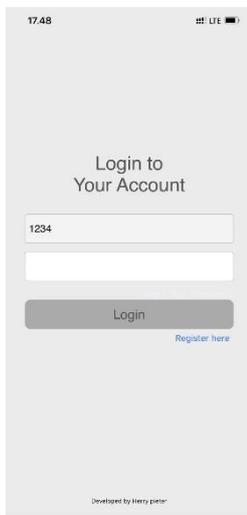
1. Mencari kalimat yang mengandung "Total", "Diskon" dan "Subtotal".
2. Menggunakan regex `"[-]{2}"`, yang bertujuan untuk mencari pembatas garis nota yang mempunyai pola "----" yang berhasil terbaca oleh Google Vision.
3. Menggunakan regex `"[a-zA-Z]:"`, yang bertujuan untuk mencari pola "Total:" yang terdapat pada bagian footer.
4. Menggunakan regex `"^([a-zA-z])$"`, yang bertujuan untuk mencari karakter salah pengenalan dari Google Vision.

Proses looping bertujuan untuk mendapatkan index kalimat yang akan digunakan sebagai batasan dalam pemotongan data array. Setelah data melalui dua proses pemotongan, maka proses terakhir yang dilakukan adalah pengambilan data barang. Dalam proses pengambilan data, akan dilakukan looping terhadap array untuk memisahkan data menjadi 4 bagian, yaitu nama, kuantiti, diskon, dan harga. Terdapat beberapa step yang dilakukan dalam pengambilan data, proses tersebut yaitu:

1. Setiap baris kalimat akan dilakukan pengecekan apa kalimat tersebut nama barang atau bukan dengan menggunakan regex `"[a-zA-Z0-9]{2} [a-zA-Z0-9]{2}"`.
2. Apabila kalimat tersebut sesuai dengan regex pada poin 1, maka kalimat tersebut akan di split dengan karakter spasi.

3. Kata yang dipecah akan dilakukan pengecekan untuk mencari apabila ada harga, kuantiti atau harga.
4. Regex yang digunakan untuk pengecekan kuaniti yaitu “[0-9]”.
5. Regex yang digunakan dalam aplikasi untuk pengecekan harga, yaitu “[1-9]{1}[\.,][0-9]{3}”.
6. Regex yang digunakan untuk melakukan pengecekan diskon yaitu “-[0-9]{2}”.
7. Apabila kalimat tidak sesuai dengan regex pada poin pertama, maka kalimat tersebut akan displit dengan pemisah karakter spasi. Kemudian, setiap kata yang merupakan hasil split akan dilakukan pengecekan dengan menggunakan pola atau regex untuk kuantiti, harga dan diskon. Apabila kata yang didapat tidak sesuai dengan pola yang sudah ditentukan, maka kata tersebut akan diabaikan.
8. Setelah semua data pada array selesai dipisahkan menjadi empat bagian, selanjutnya data pada array harga akan dinormalisasi untuk menghilangkan karakter ‘,’ dan ‘.’, yang bertujuan agar dapat diconvert menjadi tipe data Int.
9. Setelah harga dinormalisasi, maka keempat bagian yang sudah dipisahkan akan disatukan menjadi object yang akan digunakan pada aplikasi.

Proses pengambilan data yang dilakukan memiliki beberapa kelemahan, yaitu apabila produk hanya terdiri dari 1 kata, maka produk tersebut tidak dapat terbaca. Selain itu, apabila nama produk hanya mengandung angka, maka nama produk tersebut akan terbaca sebagai harga.



Gambar. 6. Halaman Login

Pada gambar 6 adalah tampilan login saat user sudah mendaftar pada aplikasi. Untuk login user perlu memasukkan email / nomor telepon dan password. User dapat menekan tombol masuk ketika email dan password terisi dengan benar. Jika pengguna salah memasukkan nomor telepon yang digunakan, maka aplikasi akan memunculkan pesan error “Nomor hp / password salah”.



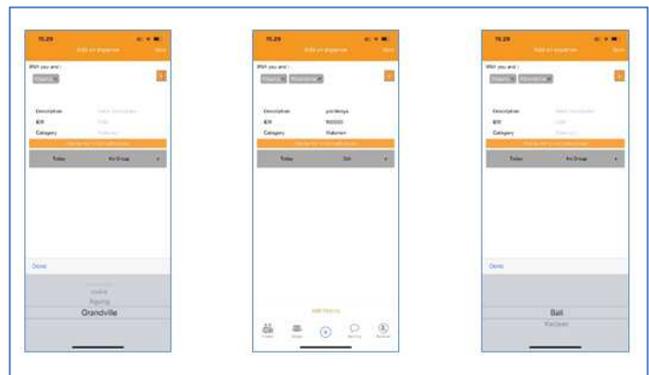
Gambar. 7. Halaman Utama

Pada gambar 7 adalah halaman daftar teman dan total hutang atau piutang dari pengguna dan juga halaman utama ketika pengguna melakukan login. Gambar 7 adalah daftar teman yang dimiliki oleh pengguna beserta total hutang dan piutang dari masing-masing teman yang ada.



Gambar. 8. Halaman Group

Pada gambar 8 adalah daftar keseluruhan group yang dimiliki oleh pengguna beserta total hutang atau piutang yang di miliki pengguna di group tersebut. Pada saat pengguna baru melakukan register dan login, maka halaman ini akan menjadi kosong.



Gambar. 9 Tambah Hutang atau piutang

Gambar 9 merupakan tampilan untuk menambahkan transaksi di dalam aplikasi manajemen hutang piutang. Pengguna dapat memilih dengan siapa saja ingin melakukan

transaksi, atau dengan group mana transaksi tersebut. Pengguna juga dapat memilih kategori, mengisi tanggal, deskripsi dan nominal transaksi.



Gambar. 10 Halaman Passcode

Pada gambar 10 merupakan halaman dari passcode pada aplikasi manajemen hutang piutang. Untuk memunculkan halaman ini, pengguna harus mengaktifkan passcode pada halaman my account. Face ID akan otomatis dapat digunakan setelah pengguna mengaktifkan passcode.



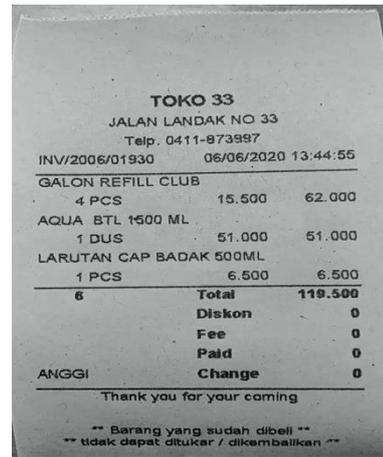
Gambar. 11 Halaman Laporan

Gambar 11 merupakan contoh laporan pada aplikasi manajemen hutang piutang. Laporan pada gambar 11 merupakan laporan total hutang yang dikelompokkan berdasarkan teman. Selain itu, terdapat juga laporan total hutang dan piutang serta laporan total piutang yang dikelompokkan berdasarkan dengan teman yang dimiliki pengguna.

V. UJI COBA

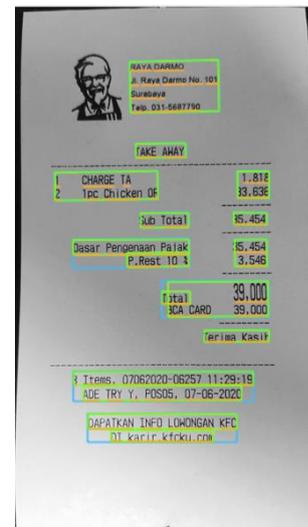
Pada tahap ini akan dijelaskan mengenai uji coba yang dilakukan pada aplikasi manajemen hutang piutang ini. Uji coba ini bertujuan untuk mengetahui semua fungsi berjalan dengan baik dan uji kelayakan bertujuan untuk mengetahui kekurangan dari calon pengguna.

Fitur yang diuji coba merupakan fitur-fitur utama di dalam aplikasi meliputi OCR, laporan, passcode, serta menambahkan hutang atau piutang pada aplikasi.



Gambar. 12. Output Preprocessing

Pada gambar 12 merupakan hasil setelah melakukan preprocessing pada gambar. Proses ini terdiri dari gray scalling, contrast stretching dan rectangle detection. Proses ini bertujuan untuk membuat gambar yang blur menjadi lebih jelas dan juga membuat output dari Google Vision menjadi lebih optimal.



Gambar. 6. Output Segmentasi Google Vision

Gambar 6 merupakan hasil uji coba segmentasi Google Vision. Pada hasil uji coba segmentasi yang dilakukan terhadap 30 nota, terdapat total 15 nota yang mempunyai akurasi kurang dari 100%, yaitu 4 nota Alfamart, 2 nota Indomaret, 7 nota Toko 33, dan 2 nota Janji Jiwa. Sehingga akurasi segmentasi yang dihasilkan Google Vision ialah 94.79%. Penyebab utama kesalahan yang terjadi ialah Google Vision tidak dapat mendeteksi satu digit angka yang posisinya berada di ujung nota. Penyebab lainnya karena Google Vision mendeteksi watermark pada nota, sehingga akan berpengaruh pada hasil pembacaan karakter yang dilakukan



Gambar. 7. Hasil Uji Coba Pengambilan Barang

Gambar 7 merupakan input dari proses pengambilan barang beserta atributnya pada aplikasi manajemen hutang piutang. Pada uji coba proses pengambilan nama barang

terhadap 30 nota. Tingkat akurasi yang dihasilkan sebesar 89.99%, dengan perbandingan jumlah nama, kuantitas, dan harga yang berhasil diambil dibagi dengan jumlah nama, kuantitas, dan harga keseluruhan yang terdapat pada nota. Penyebab kesalahan yang terjadi yaitu :

1. Pada saat proses pengambilan nama beserta atributnya karena kesalahan posisi dari atribut barang yang terdapat di bagian footer.
2. Data yang diterima masih terdapat footer dan header, sehingga data tersebut dibaca menjadi data barang beserta atributnya.
3. Nama barang pada nota berpindah menjadi dua baris, yang menyebabkan aplikasi manajemen hutang piutang mengganggu dua produk yang berbeda.

Kuisisioner ini akan dijelaskan hasil kuisisioner yang dibagikan kepada minimal lima belas pengguna sesuai kriteria pada proposal yaitu usia 17 sampai 40 tahun. Terdapat lima belas responden yang menggunakan aplikasi minimal tiga kali kemudian menjawab kuisisioner yang diberikan.

Tabel 1
Pertanyaan dan Presentase Jawaban Untuk Aplikasi Yang Dibuat

No	Pertanyaan	Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
1	Seberapa mudah menggunakan aplikasi ini?	4,2% (0 orang)	0% (0 orang)	33,33% (5 orang)	66,67% (10 orang)	0% (0 orang)
2	Apakah aplikasi ini mempermudah dalam proses pencatatan hutang dan piutang?	0% (0 orang)	26,6% (4 orang)	46,6% (7 orang)	26,8% (4 orang)	0% (0 orang)
3	Apakah fitur aplikasi sudah cukup ?	0% (0 orang)	13,3% (2 orang)	66,7% (10 orang)	20% (3 orang)	0% (0 orang)
4	Bagaimana dengan tampilan keseluruhan aplikasi ?	0% (0 orang)	46,7% (7 orang)	46,7% (7 orang)	6,6% (1 orang)	0% (0 orang)
5	Seberapa puas menggunakan aplikasi ?	0% (0 orang)	26,7% (4 orang)	53,4% (8 orang)	19,9% (3 orang)	0% (0 orang)

Keterangan hasil yang dirinci pada tabel 1 membuktikan bahwa kenyamanan user untuk pemakaian aplikasi cukup baik. Persentase kemudahan aplikasi memiliki penilaian yang lebih tinggi dibanding penilaian buruk. Kemudian presentasi kepuasan responden dalam menggunakan aplikasi sudah cukup baik karena memiliki nilai yang lebih tinggi dibanding pengguna yang kurang puas.

VI. KESIMPULAN

Pada subbab ini akan dijelaskan mengenai kesimpulan yang didapat selama mengerjakan tugas akhir. Adapun beberapa proses pengerjaan seperti penggunaan swift language, penggunaan firebase, pengguna web service, alur sistem dan semua masalah yang dihadapi semasa

pengerjaan. Berikut ini penjabaran kesimpulan yang didapatkan dalam bentuk poin-poin:

- 1) Aplikasi sudah cukup memenuhi kebutuhan pengguna dalam pencatatan hutang dan piutang, dengan presentasi 26.6% menyatakan kurang, 46,6% menyatakan cukup dan 26,8% menyatakan baik.
- 2) Pengerjaan aplikasi manajemen hutang piutang menggunakan Swift cukup mempermudah dalam pengerjaan aplikasi manajemen hutang piutang. Selain itu, Swift juga mempunyai banyak plugin bawaan seperti charts, json decoder, alamofire yang membantu dalam pengerjaan aplikasi manajemen hutang piutang. Selain itu banyak library open source yang tersedia dan sangat membantu dalam proses pengerjaan aplikasi manajemen hutang piutang.
- 3) Penggunaan storyboard sangat mempermudah dalam melihat alur program, tetapi sangat sulit untuk melakukan design UI/UX yang bagus karena keterbatasan komponen dan bahasa yang sangat cepat berubah, sehingga mempersulit dalam hal desain tampilan.
- 4) Penggunaan bahasa pemrograman PHP untuk web service sudah cukup membantu dalam pembuatan web service. Namun slim framework masih kurang baik dalam menerapkan *object oriented* dan *eloquent*, sehingga banyak proses yang dilakukan.
- 5) Tingkat keberhasilan segmentasi Google Vision ialah 94.79%.
- 6) Tingkat akurasi pengenalan nama beserta harga dan kuantitas produk oleh Google Vision sebesar 93.72%.
- 7) Tingkat akurasi proses pemotongan header pada aplikasi manajemen hutang piutang sebesar 97.99%.
- 8) Tingkat akurasi proses pemotongan footer pada aplikasi manajemen hutang piutang sebesar 94.62%.
- 9) Tingkat akurasi pengambilan nama barang beserta kuantitas dan harga terhadap 30 nota sebesar 89.99%.
- 10) Rata-rata tingkat akurasi dari proses ocr hingga pengambilan nama barang agar dapat digunakan sebesar 94.2%
- 11) Implementasi cekmutasi.co.id untuk OVO, BCA, dan Mandiri memiliki proses yang sama. Namun, untuk menggunakan OVO mempunyai tingkat bahaya lebih tinggi, yang disebabkan adanya fitur transfer yang disediakan.

DAFTAR PUSTAKA

- [1] G. Nagy, T. A. Nartker, and S. V Rice, "Optical character recognition: An illustrated guide to the frontier," in *Document Recognition and Retrieval VII*, 1999, vol. 3967, pp. 58–69.
- [2] L. Moroney, Moroney, and Anglin, *Definitive Guide to Firebase*. Springer, 2017.
- [3] M. A. Mekar, S. O. Fageeri, and S. E. Fattoh, "Using firebase cloud messaging to control mobile applications," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCEEE)*, 2019, pp. 1–5.
- [4] M. Mathias and J. Gallagher, *Swift Programming: The Big Nerd Ranch Guide*. Pearson Technology Group, 2016.
- [5] D. K. Barry, *Web services, service-oriented architectures, and cloud computing*. Elsevier, 2003.
- [6] A. Sunardi and others, "MVC architecture: A comparative study

- between laravel framework and slim framework in freelancer project monitoring system web based,” *Procedia Comput. Sci.*, vol. 157, pp. 134–141, 2019.
- [7] D. Mulfari, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, “Using Google Cloud Vision in assistive technology scenarios,” in *2016 IEEE symposium on computers and communication (ISCC)*, 2016, pp. 214–219.
- [8] A. Bud, “Facing the future: The impact of Apple FaceID,” *Biometric Technol. today*, vol. 2018, no. 1, pp. 5–7, 2018.
- [9] J. Chaki and N. Dey, *A beginner’s guide to image preprocessing techniques*. CRC Press, 2018.
- [10] A. S. B. Nugroho and M. S. Al Ihsan, “Aplikasi Startup Dokter Sepatu Menggunakan Pembayaran Online Realtime Berbasis Web Dan Android,” in *Seminar Nasional Riset Terapan*, 2019, vol. 4, pp. A25--A31.