

# RANCANG BANGUN SISTEM PEMINDAIAN STRUK BELANJA UNTUK MENDAPATKAN RINCIAN BELANJA

Yuli Astuti<sup>1)</sup>, Kuswandanu Kusuma Wicaksana<sup>2)</sup>

<sup>1,2)</sup> Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta  
Jl. Ring Road Utara, Condongcatur, Depok, Sleman, Yogyakarta 55283  
Email : [yuli@amikom.ac.id](mailto:yuli@amikom.ac.id)<sup>1)</sup>, [kuswandanu.w@students.amikom.ac.id](mailto:kuswandanu.w@students.amikom.ac.id)<sup>2)</sup>

## Abstrak

*Struk bukan hanya sebagai bukti dari suatu pembelian. Sebagian orang menyimpan struk untuk dicatat isinya dengan tujuan dapat digunakan sebagai referensi pembelian di kemudian hari. Mencatat isi struk di kertas sangatlah tidak efisien, apalagi menyimpan kertasnya.*

*Penelitian ini mencoba membuat sistem yang dapat melakukan pencatatan isi struk hanya berdasarkan foto struk. Struk yang digunakan untuk percobaan adalah struk dari Indomaret dan Alfamart karena dengan banyaknya toko memiliki format struk yang hampir sama. Teknologi utama yang digunakan adalah OCR (Optical Recognition System) yang dapat mengambil isi teks dari suatu gambar. Setelah didapatkan teks dari gambar tersebut maka teks tersebut diolah untuk diambil informasinya.*

*Hasil dari penelitian ini adalah sistem yang dibuat mampu membaca isi struk dengan baik. Faktor mempengaruhi baik buruk struk adalah kualitas hasil foto. Dari 74 hasil percobaan terhadap berbagai gambar struk Indomaret dan Alfamart didapatkan hasil rata-rata mencapai akurasi 75%.*

**Kata kunci:** Computer Vision, OCR, Regular Expression, Pemindaian Struk.

## 1. Pendahuluan

### 1.1 Latar Belakang

Meskipun kadang dianggap tidak berguna, beberapa orang menyimpan *struk* belanja untuk digunakan di kemudian hari sebagai pembandingan harga suatu barang. Penyimpanan *struk* sebenarnya bisa dilakukan dengan memfotonya melalui *gadget*. Namun penyimpanan gambar *struk* dari hasil foto menyulitkan jika ingin membuka dan mencarinya lagi. Karena pencarian berkas gambar biasanya hanya memuat nama berkas atau tanggal pengambilan. Sehingga akan sulit jika ingin mencari gambar *struk* untuk pembelian barang tertentu atau parameter tertentu lainnya.

Bukan hanya disimpan saja, kadang orang juga mencatat pengeluaran berdasarkan isi dari *struk* belanja. Setiap rincian dari *struk* ditulis terutama harga setiap barang. Pencatatan manual seperti itu memakan waktu dan terkesan menjenuhkan jika yang dicatat terlalu banyak.

Meskipun pencatatan bisa dilakukan dengan media digital, tetap saja untuk memasukkan data merupakan hal yang sangat membosankan. Apalagi jika aplikasi yang digunakan kurang restruktur dalam hal manajemen penyimpanan data.

### 1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sistem yang dapat melakukan pembacaan isi *struk* belanja secara otomatis dari gambar *struk* sehingga didapatkan informasi rincian belanja. Sistem yang dibuat mencakup pada hal-hal berikut:

- 1) Sistem yang akan dibuat berupa *web service* yang terhubung dengan *Azure Cloud Service* dan diaplikasikan pada sisi *client* dengan platform *Android*.
- 2) Perangkat yang digunakan minimal memiliki versi *Android* 4.0.3 dan maksimal 7.0.
- 3) Gambar *struk* memiliki minimal resolusi 5 *Megapixel* dengan kualitas bagus, yaitu dapat dibaca dengan mudah olah manusia, tidak *blur*, tidak banyak *noise*, memiliki pencahayaan yang merata, dan tidak berkerut.
- 4) *Struk* yang akan digunakan adalah *struk* belanja dari minimarket *Indomaret* dan *Alfamart* yang memiliki berbagai cabang sehingga bentuk dan susunan *struk* tidak terlalu bervariasi.
- 5) *Struk* belanja memiliki jumlah baris tulisan tidak lebih dari 40 baris dan maksimal rasio antara lebar dan panjang *struk* adalah 2:5.

### 1.3 Tinjauan Pustaka

Penelitian tentang bagaimana membaca *struk* belanja secara otomatis bukanlah hal yang baru. Sebelumnya sudah ada penelitian serupa, namun belum dijumpai di Indonesia. Salah satunya dilakukan oleh Janssen, dkk. yang membuat sistem bernama *Receipts2Go*. Pada penelitian tersebut disimpulkan *Receipt2Go* berhasil mengambil informasi dari berbagai macam jenis *struk* tanpa mengetahui format tata letak *struk* sebelumnya. Namun yang diambil hanya beberapa informasi seperti nama toko, tanggal pembelian dan total harga. Sedangkan rincian masing-masing barang yang dibeli tidak diperhatikan. Tahapan yang dilakukan adalah *Image Normalization* yang terdiri dari koreksi kemiringan gambar dan menghapus *noise*. Kemudian

melakukan *Entity Extraction* yaitu pembacaan karakter menggunakan sistem OCR serta menerapkan *regular-expression* untuk menemukan pola tertentu seperti format tanggal (23/10/16) dan suatu kata atau frase tertentu seperti "DATE PAID". Dan terakhir melakukan *Information Extraction* yang menarik kesimpulan berdasarkan pola-pola yang diperoleh sebelumnya [1].

Ozarslan, dkk. mengusulkan metode *Knowledge Based Correction* (KBC) serta *Row Based Multiple Frame Integration* (RB-MFI) [2]. Metode KBC digunakan untuk melakukan koreksi terhadap hasil dari proses OCR dari *Tesseract OCR Engine*. Pengecekan kesalahan berdasarkan pada data yang sudah dimiliki. Hasilnya, akurasi pengenalan karakter dan kata meningkat sampai 20% jika dibandingkan dengan tanpa KBC. Sedangkan Metode RB-MFI merupakan metode untuk proses akuisisi dari video sehingga didapatkan beberapa *frame*. Dengan begitu akan didapatkan beberapa kandidat citra yang bisa diproses. Dari hasil yang didapat akurasi pengenalan kata meningkat 26% dan pengenalan karakter meningkat 25% [3].

## 2. Pembahasan

### 2.1 Analisis Masalah

Analisis dilakukan terhadap contoh *struk*. Gambar 1 menunjukkan contoh *struk* yang dianalisis, *struk* tersebut mewakili *struk* yang lain karena formatnya hampir sama. Pengamatan terhadap *struk* dapat diambil kesimpulan bahwa *layout* (tata letak) setiap *struk* dapat dibagi menjadi 3 bagian utama, yaitu *header*, *body* dan *footer*.



Gambar 1. Bagian-bagian struk

#### 1) Header

Seperti yang ditunjukkan pada warna merah, bagian *header* berisi informasi minimarket. Informasi tersebut berupa nama minimarket, nama perusahaan, nama cabang, alamat, nomor telepon, serta ada yang menyertakan NPWP. Pada *struk Indomaret* yang menunjukkan itu *Indomaret* adalah logonya. Sedangkan pada *Alfamart* tidak ada kata *Alfamart* melainkan nama perusahaan baik *Alfamart* maupun waralaba. Selain itu

pada bagian ini juga memuat nomor transaksi dan nama operator kasir. Untuk yang *Indomaret* waktu transaksi ditulis juga pada bagian ini.

#### 2) Body

ULTRA SLIM COKLAT200	1	4600	4,600
SARI ROTI CKT KJU 72	1	5500	5,500
PLASTIK SDG	1	1	1
DISKON :			(1)
POCARI SWEAT 50	1	7,100	7,100
WALL' PP FRUIT	1	3,200	3,200
KANTONG PLS M	1	1	1
Disc.	-1		

Gambar 2. Bagian Daftar Barang pada Body

Bagian ini berisi rincian pembelian. Intinya memuat daftar barang apa saja yang dibeli, jumlah barang, harga setiap masing-masing barang dan harga total sesuai jumlah pembelian. Keempat elemen tersebut bisa dipisah dengan dijadikan kolom-kolom yang rata seperti pada Gambar 2. Kolom pertama yang berwarna merah memuat nama barang yang ditulis tidak utuh melainkan disingkat. Contoh nama barang adalah "SARI ROTI CKT KJU 72" yang ternyata kata-kata tidak semua ditulis melainkan disingkat. Seperti "CKT" yang kemungkinan besar maksudnya cokelat dan "KJU" yang maksudnya keju. Ada juga "KANTONG PLS M" maksudnya kantong plastik berukuran medium atau sedang.

Kemudian kolom berwarna kuning merupakan kolom untuk jumlah pembelian tiap barang. Kolom berwarna hijau adalah harga setiap satu barang. Dan paling kanan berwarna biru adalah total harga masing-masing jenis barang yang merupakan perkalian dari kolom berwarna kuning dan kolom berwarna hijau.

Juga perlu diperhatikan di sana adalah adanya baris diskon. Baris tersebut sebenarnya hampir sama dengan baris-baris di atasnya namun untuk *Indomaret* terletak lebih menjorok ke kanan dan harganya tertulis dengan tutup kurung "(1)". Sedangkan pada *struk Alfamart* letaknya sejajar vertikal dengan nama barang di atas namun harganya tidak berada di kanan melainkan lebih ke kiri satu kolom dengan nama barang namun rata kanan, serta ditulis dengan tanda negatif "-1".

#### 3) Footer

Yang terakhir yang diberi warna biru berisi informasi tambahan. *Struk Alfamart* menaruh waktu di bagian ini. Di paling bawah terdapat informasi alamat *web* dari keduanya yaitu "WWW.INDOMARET.CO.ID/PROMO" dan "WWW.ALFAMART.COM". Dan selain itu juga terdapat kalimat yang mengandung kata *Indomaret* dan *Alfamart*. Kedua hal ini juga membantu untuk mencari tahu nama toko.

2.2 Solusi Masalah

2.2.1 Pembacaan Teks

Proses ini adalah mengambil isi teks dari gambar *struk*. Untuk melakukannya digunakan layanan *OCR* dari *Microsoft Azure Cognitive Service*.

2.2.2 Penggalan Informasi

- 1) Mendapatkan Nama Toko

Untuk mendapatkan nama toko bisa ditentukan dengan mencocokkan pencarian kata toko, yaitu “*Indomaret*” atau “*Alfamart*”. Pencarian hanya dilakukan pada baris *header* dan *footer*. Selain menggunakan 2 kata tersebut pencarian juga ditambah dengan alamat *website* “*WWW.INDOMARET.CO.ID/PROMO*” dan “*WWW.ALFAMARTKU.COM*”.

- 2) Mendapatkan Tanggal Transaksi

Untuk mendapatkan tanggal transaksi dilakukan pencarian dengan pola menggunakan *Regex*. Contoh tanggal adalah *dd-mm-yyyy* atau *dd.mm.yyyy*. Pencarian juga hanya dilakukan pada semua teks.

- 3) Mendapatkan Daftar Barang Beserta Harganya

Untuk mendapatkan daftar barang beserta harganya maka dilakukan pencarian hanya pada bagian *body* saja. Pencarian dengan memilah kolom yang dipisah dengan spasi. Dimulai dari kolom paling kanan adalah harga total, kolom kirinya adalah harga per barang, kemudian harga kirinya lagi adalah jumlah barang yang dibeli, dan kolom paling kiri adalah nama barang. Pencarian dimulai dari kanan karena pada kolom paling kiri yang berisi nama barang tidak hanya terdiri dari 1 kata. Pencarian untuk daftar barang hanya dilakukan hingga baris sebelum ada kata-kata yang menunjukkan total harga. Kata-kata tersebut seperti “*TOTAL HARGA*”, “*TOTAL ITEM*”, “*HARGA JUAL*”, dan lain-lain.

- 4) Mendapatkan Total Harga

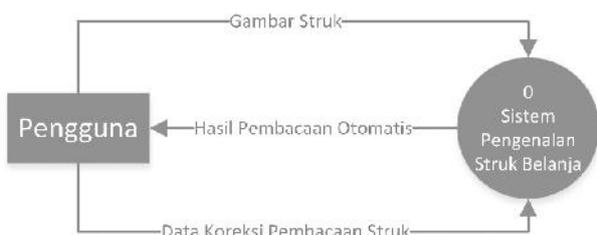
Seperti pada saat mendapatkan harga barang maka untuk mendapatkan total harga dilihat baris yang memiliki kata-kata yang menunjukkan total harga.

2.3 Perancangan Sistem

2.3.1 Pemodelan Proses

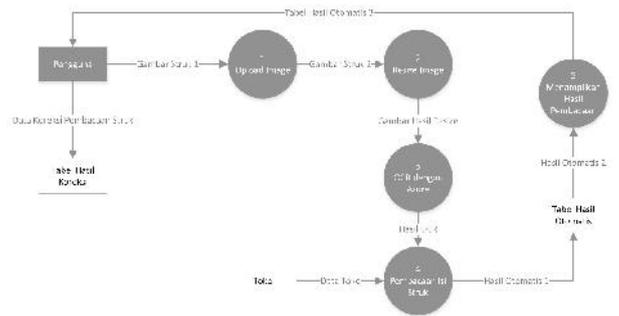
2.3.1.1 Data Flow Diagram (DFD)

- 1) Context Diagram



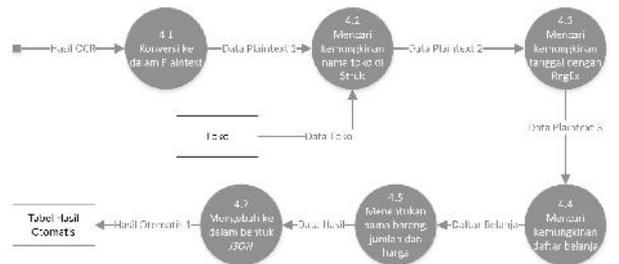
Gambar 3. Context Diagram

- 2) DFD Level 0



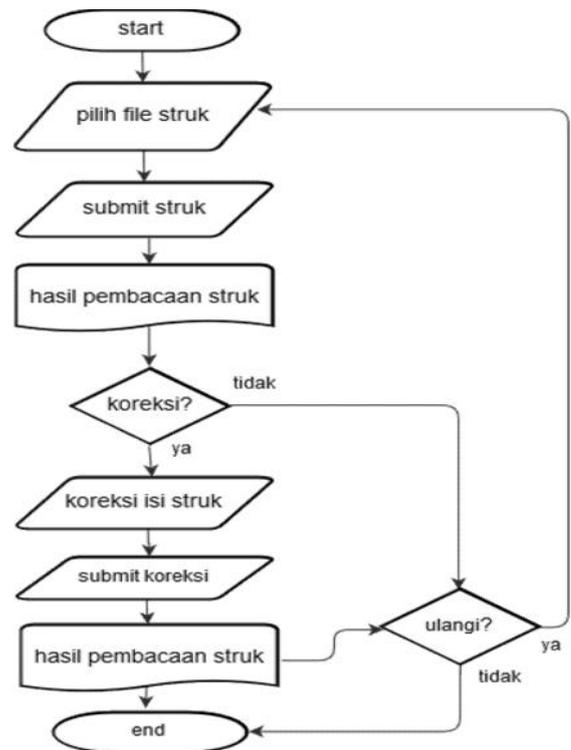
Gambar 4. DFD Level 0

- 3) DFD Level 1 Proses 4



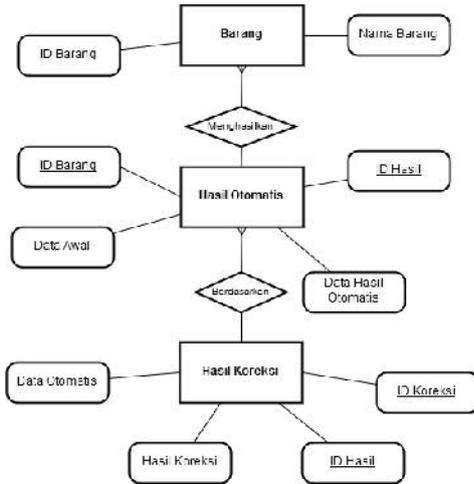
Gambar 5. DFD Level 1 Proses 4

2.3.1.2 Flowchart



Gambar 6. Proses Penggunaan Aplikasi

2.3.2 Pemodelan Data dengan ERD



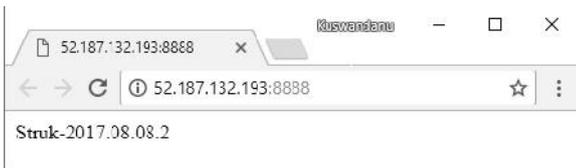
Gambar 7. ERD

2.4 Implementasi Sistem

Pada bagian ini ada 2 tahap, yaitu pembuatan *server* dan program *client*.

2.4.1 Program Server

Untuk program *server* menggunakan *web server* sederhana yaitu Web.py dengan bahasa pemrograman Python.



Gambar 8. Akses Server

2.4.2 Program Client

Program *client* berupa aplikasi *Android* yang berhubungan dengan *server*

1) Tampilan Awal



Gambar 9. Tampilan Awal Client

Pada tampilan awal terdapat beberapa hasil pembacaan.

2) Memilih Gambar

Memilih gambar bisa dari kamera langsung atau memilih dari galeri.

3) Memproses Gambar

Pada tahap ini gambar diupload ke server kemudian menunggu respons dalam format JSON yang kemudian diolah



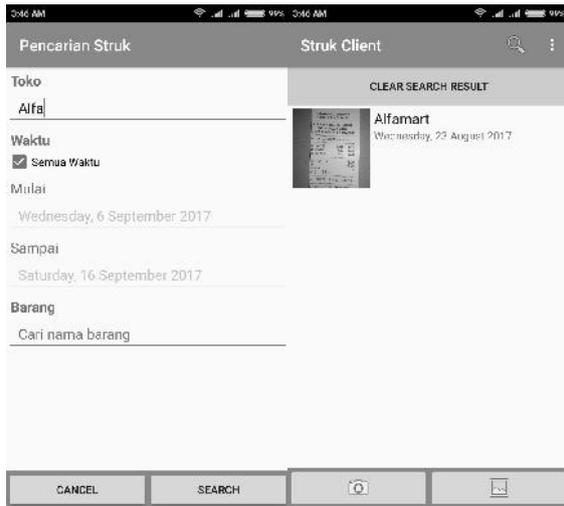
Gambar 10. Memproses Gambar

4) Hasil dan Koreksi



Gambar 11. Hasil dan Koreksi

5) Pencarian Struk



Gambar 12. Pencarian

2.5 Hasil Pengujian

Pengujian dilakukan pada beberapa *struk* yang ditangkap gambar menggunakan kamera pada perangkat seperti Microsoft Lumia 535 (5 MP), Asus Zenfone Go ZB 500KL (12 MP) dan Xiaomi Redmi 4A (12 MP). Pengujian dilakukan pada 74 gambar *struk*. Dari hasil yang didapat, tingkat akurasi di atas 60% dengan rata-rata dari semua hasil percobaan adalah sekitar 75% akurasi. Akurasi dihitung menggunakan rumus:

$$a = \frac{v_t + v_w + \sum_0^b (v_n + v_h + v_j)}{2 + 3b} \cdot 100\% \dots\dots (1)$$

Dengan *a* = akurasi, *t* = toko, *w* = waktu, *b* = banyak barang, *n* = nama, *h* = harga, *j* = jumlah. Sementara *v* adalah nilai dari masing-masing variabel, jika benar nilainya 1 dan jika salah nilainya 0.

Tabel 1. Perbandingan Hasil Beda Perangkat

Prangkt	Gambar dan Hasil	Akrs (%)
Microsoft Lumia 535 kamera 5 Megapixel		62.5

	<pre>{   "belanja": [     {       "harga": 6500,       "jumlah": 0,       "nama": "FRESTEA GRN MADU"     },     {       "harga": 8500,       "jumlah": 0,       "nama": "CIMORY YGRT BLBRY250"     }   ],   "status": 1,   "toko": "indomaret",   "total": 0,   "waktu": "2017-09-17" }</pre>	
Xiaomi Redmi 4A kamera 12 Megapixel		87.5
	<pre>{   "belanja": [     {       "harga": 6500,       "jumlah": 1,       "nama": "FRESTEA GRN MADU"     },     {       "harga": 8500,       "jumlah": 1,       "nama": "CIMORY YGRT BLBRY250"     }   ],   "status": 1,   "toko": "indomaret",   "total": 15000,   "waktu": "2017-09-17" }</pre>	

Dalam pengujian juga dibandingkan perbedaan antara hasil pembacaan gambar dari perangkat berbeda jenis serta dengan pencahayaan yang berbeda. Untuk pengujian beda perangkat dibandingkan antara Lumia 535 dan Redmi 4A yang berbeda resolusi kameranya. Sedangkan untuk perbandingan pencahayaan digunakan perangkat yang sama yaitu Redmi 4A dengan perbedaan menggunakan *flash* dan tanpa *flash*.

Dari hasil yang didapatkan pengujian dengan perbedaan perangkat menunjukkan bahwa adanya perbedaan hasil akhir yang mempengaruhi nilai akurasi. Perbedaan yang terlihat adalah hanya pada kolom jumlah barang. Hasil dari Lumia 535 tidak dapat membaca jumlah barang. Hal ini menunjukkan perbedaan kualitas kamera mempengaruhi hasil pembacaan.

Sedangkan untuk pencahayaan, perbedaan juga sedikit terjadi meskipun hasil akhir akurasi sama karena kesalahan saat pemilahan. Hasil tanpa *flash* bahkan salah membaca total harga tiap item.

Tabel 2. Hasil Perbandingan Pencapaian

Flash	Gambar dan Hasil	Akrs (%)
Ya	 <pre>                     {                     "belanja": [                     {                     "harga": 5200,                     "jumlah": 1,                     "nama": "AQUA AIR MINERAL"                     },                     {                     "harga": 4800,                     "jumlah": 2,                     "nama": "BENG-BENG CHOCO 22G"                     }                     ],                     "status": 0,                     "toko": "Indomaret",                     "total": 14800,                     "waktu": "2017 05 19"                     }                 </pre>	87.5
Tidak	 <pre>                     {                     "belanja": [                     {                     "harga": 5200,                     "jumlah": 1,                     "nama": "AQUA AIR MINERAL"                     },                     {                     "harga": 800,                     "jumlah": 2,                     "nama": "BENG-BENG CHOCO 22G"                     }                     ],                     "status": 1,                     "toko": "indomaret",                     "total": 6800,                     "waktu": "2017-05-19"                     }                 </pre>	87.5

- 1) Pembuatan sistem yang mampu membaca isi dari gambar *struk* secara otomatis berhasil dilakukan dengan menggunakan bantuan dari layanan *Microsoft Azure OCR*.
- 2) Implementasi pada perangkat *client Android* juga berhasil. Aplikasi dapat mengambil gambar *struk* kemudian mengirim gambar ke *server* yang nantinya didapatkan hasil pembacaan isi *struk* otomatis.
- 3) *Struk* dapat dibaca dengan baik jika gambar jelas dan bentuk *struk* tidak melekek serta pencahayaan merata.
- 4) Hasil pemilahan data daftar belanja ada beberapa yang tidak akurat, hal ini dikarenakan ketidakcocokan dengan aturan pemilahan.
- 5) Nama barang tidak didapat nama asli barang yang lengkap. Hal ini karena pada *struk*, nama barang ditulis pendek bahkan hanya singkatan.

**Daftar Pustaka**

[1] B. Janssen, E. Saund, E. Bier, P. Wall dan M. A. Sprague, "Receipts2Go: The Big World of Small Documents," dalam *ACM Symposium on Document Engineering*, Paris, France, 2012.

[2] S. Ozarslan dan P. E. Eren, "Text recognition and correction for automated data collection by mobile devices," dalam *Proc. SPIE 9027, Imaging and Multimedia Analytics in a Web and Mobile World*, San Francisco, California, United States, 2014.

[3] S. Ozarslan dan P. E. Eren, "Multi-frame knowledge based text enhancement for mobile phone captured videos," dalam *Proc. SPIE 9030, Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications*, San Francisco, California, United States, 2014.

**Biodata Penulis**

*Yuli Astuti*, memperoleh gelar Magister Komputer (M.Kom) dari Universitas AMIKOM Yogyakarta pada tahun 2012. Saat ini menjadi Dosen di Universitas AMIKOM Yogyakarta.

*Kuswandanu Kusuma Wicaksana*, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Informatika Universitas AMIKOM Yogyakarta, lulus tahun 2017.

**3. Penutup**

**3.1 Kesimpulan**

Dari hasil pengujian aplikasi menggunakan beberapa *struk*, penulis dapat menarik beberapa kesimpulan: