

---

# IMPLEMENTASI ALGORITMA *FLOYD WARSHALL* DALAM MENENTUKAN JALUR TERBAIK *DRIVER PASTIFRESH.ID*

Adiston Sitompul<sup>1</sup>, Jeri Charles<sup>2</sup>, Daniel Udjulawa<sup>3</sup>

STMIK GI MDP; JL. Rajawali No.14 Palembang, 0711-376400

Jurusan Teknik Informatika, STMIK GI MDP, Palembang

e-mail: <sup>1</sup>[jericharles@mhs.mdp.ac.id](mailto:jericharles@mhs.mdp.ac.id), <sup>2</sup>[adistonsitompul@mhs.mdp.ac.id](mailto:adistonsitompul@mhs.mdp.ac.id),  
<sup>3</sup>[danieludjulawa@mdp.ac.id](mailto:danieludjulawa@mdp.ac.id)

## **Abstrak**

*Pastifresh.id* adalah sebuah perusahaan teknologi dari indonesia yang melayani pesan antar bahan untuk kebutuhan dapur secara online. Sebuah pencarian jarak terpendek sangat dibutuhkan oleh driver *pastifresh.id*, karena dalam melakukan pengiriman juga membutuhkan sebuah solusi mengenai rute yang akan dilewati haruslah jarak yang paling minimum dari posisi vendor yang terkait kepada konsumen. Tujuan penelitian adalah untuk mengimplementasikan algoritma Floyd Warshall dalam menentukan jalur terbaik driver *Pastifresh.id*. Algoritma Floyd Warshall adalah penentuan jalur terpendek dengan menggunakan graf berbobot untuk menghasilkan jalur terpendek. Perhitungan algoritma Floyd Warshall dengan menghitung bobot terkecil dari titik awal ke titik tujuan. Hasil perhitungan akan diterapkan kedalam perangkat lunak aplikasi dengan menggunakan Android Studio. Analisis dan kebutuhan aplikasi bagi pengguna akan dikembangkan dengan metodologi prototyping. Hasil pengujian dari Algoritma Floyd warshall dapat mengoptimalkan matriks jarak untuk menemukan rute terpendek dari *pastifresh.id* ke customer dengan melewati semua titik vendor yang telah ditentukan.

**Kata Kunci:** algoritma Floyd Warshall, Jalur Terpendek, Android Studio, Prototyping

## **Abstract**

*Pastifresh.id* is a technology company from indonesia that serves delievery for your kitchen needs online. a shortest distance search is needed by *pastifresh.id* driver, because in the delivery also requires a solution about the route to be passed must be the most minimum distance from the vendor-related position to the consumer. The purpose of this study is to implement Floyd Warshall algorithm in determining the best path for *Pastifresh.id* driver, Floyd Warshal Algorithm is about determining the shortest path by using a weighted graph to produce the shortest path. Floyd Warshall algorithm calculations are performed by calculating the smallest weights from the starting point to the destination point. The calculation results will be applied to the application software using Android studio. The analysis and application requirements for users will be developed with prototyping methodologies. Test results from the Floyd warshall Algorithm can optimize the distance matrix to find the shortest route from *pastifresh.id* to the customer by passing through all vendor point that has been determined.

**Keywords:** Floyd Warshall Algorithm, Shortest Path , Android Studio, Prototyping.

---

---

## 1. PENDAHULUAN

Pastifresh.id merupakan perusahaan teknologi asal Indonesia yang bergerak dibidang layanan pesan antar kebutuhan dapur online berupa bahan pokok. Konsumen bisa memesan produk melalui aplikasi dengan menyertakan alamat pengantaran. Pesanan produk konsumen akan dibuat menjadi sebuah order dengan menghubungkan produk dengan vendor yang menjualnya secara otomatis. Pesanan yang telah dibuat akan diproses oleh driver dengan cara mengambil produk dari vendor dan mengatarnya ke konsumen..

Algoritma Floyd Warshall menghitung jalur terpendek antara semua simpul dengan menghitung dari satu sumber simpul sampai simpul tujuan melalui beberapa jalur. Algoritma Floyd Warshall dapat digunakan untuk mencari panjang lintasan terpendek antara semua pasangan simpul dalam graf sederhana yang terhubung. Cara kerja dari algoritma Floyd Warshall adalah dengan membandingkan semua lintasan yang mungkin terjadi dalam graf untuk setiap pasang simpul dan melakukan pengujian dari setiap kombinasi simpul yang diperoleh.

Pencarian jarak terpendek merupakan suatu permasalahan yang sedang dialami Pastifresh saat ini, karena dalam melakukan sebuah perjalanan membutuhkan solusi bagaimana rute yang akan dilalui adalah rute atau jarak yang paling minimum (terkecil) dari vendor - vendor yang terkait ke konsumen.

Penelitian terdahulu mengenai pencarian jarak terpendek yang menggunakan algoritma Floyd Warshall yaitu Irfan Ardiansyah dan Dimara Kusuma Hakim tahun 2012 dengan judul “ Rancang Bangun Aplikasi Untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di Lokasi Wisata Purbalingga ” pada penelitian ini membahas tentang pembuatan aplikasi untuk menentukan jalur terpendek menggunakan algoritma Floyd di lokasi wisata Purbalingga. Obyek wisata besar yang ada di Kabupaten Purbalingga adalah Owabong, Akuarium Raksasa, Taman Reptil, Monumen Jendral Sudirman, dan Goa Lawa. Data jalan digunakan untuk mengetahui tujuan dan jalur yang akan dilalui untuk sampai ke obyek wisata tersebut. Satuan yang digunakan dalam jarak jalur ini adalah meter. Kesimpulan dari penelitian ini adalah dihasilkan aplikasi pencarian jalur terpendek menggunakan Algoritma Floyd yang dapat digunakan di kabupaten Purbalingga.

Penelitian terdahulu mengenai pencarian jarak terpendek yang menggunakan algoritma Floyd Warshall yaitu Anik Andriani di tahun 2014 dengan judul “ Rancang Bangun Sistem Informasi Rute Wisata Terpendek Berbasis Algoritma Floyd-Warshall ” tujuan dari penelitian ini adalah membangun Sistem Informasi berbasis web yang memberikan informasi kepada wisatawan mengenai jalur terpendek menuju obyek wisata di Yogyakarta. Algoritma Floyd Warshall dapat diterapkan dalam pencarian rute terpendek menuju obyek-obyek wisata populer di Yogyakarta. Hasilnya dapat diterapkan dalam pembangunan sistem informasi untuk rute wisata terpendek yang berbasis website.

Penelitian terdahulu mengenai pencarian jarak terpendek yang menggunakan algoritma Floyd Warshall yaitu Lutfi Fanani, Eriq M. Adams J., Satrio A. Wicaksono tahun 2012 dengan judul “ Rancang Bangun Aplikasi Web Pencarian Rute Terpendek Antar Gedung di Kampus Menggunakan Algoritma Floyd-Warshall ” Penelitian ini membahas tentang sebuah sistem informasi web pencarian rute terpendek antar gedung di kampus yang dapat memudahkan masyarakat dalam mencari sebuah fakultas atau jurusan tertentu dan jalur yang dapat dilalui untuk menuju fakultas atau jurusan dengan menggunakan algoritma Floyd Warshall. Hasil pengujian menunjukkan bahwa keseluruhan fungsional bekerja dengan baik dengan tingkat akurasi algoritma Floyd-warshall selalu menunjukkan nilai 100%.

Penelitian yang sudah dilakukan oleh peneliti terdahulu seperti di atas, belum ada yang melakukan implementasi algoritma Floyd Warshall untuk menentukan jalur terbaik Pastifresh.id.

---

Dengan alasan tersebut maka penelitian ini akan dilakukan dengan implementasi algoritma floyd warshall dalam menentukan jalur terbaik driver pastifresh.id sekaligus lintasan.

## 2. METODE PENELITIAN

Metodologi yang digunakan dalam pengembangan aplikasi ini adalah *Prototyping*. Dimulai dari analisis kebutuhan, membangun *prototype*, evaluasi *prototype*, penulisan kode program, pengujian program sampai tahap pemeliharaan semuanya menggunakan metodologi ini. *Prototyping* membagi tahapan pengembangan perangkat lunaknya ke dalam 5 tahap sebagai berikut :

### 2.1 Analisis Kebutuhan

Pada tahap ini penulis merumuskan masalah yang ada di pastifresh.id dan mengumpulkan semua kebutuhan yang diperlukan dengan membaca buku dan jurnal yang berkaitan dengan algoritma floyd warshall, android dan teori teori lain yang terkait. Serta penulis telah menganalisa penelitian-penelitian dan jurnal terdahulu serta membandingkannya untuk mempermudah penulis dalam mendapatkan kesimpulan aplikasi yang akan dibuat.

### 2.2 Membangun *Prototype*

Tahap ini dimulai dengan membuat gambaran aplikasi yang ingin dibuat, alur jalannya aplikasi dari awal hingga akhir. Dalam pembuatan desain menggunakan bahasa pemrograman yang dapat dimengerti oleh mesin yaitu handphone (*Smartphone*) yang berbasis Android melalui proses *coding*.

### 2.3 Evaluasi *Prototype*

Pada tahap ini dilakukan evaluasi untuk mengetahui apakah rancangan aplikasi yang dibuat sesuai dengan yang diharapkan. Jika tidak, maka diperbaiki dengan perubahan desain dan prototipe. Apabila aplikasi sesuai dengan yang diharapkan, maka pengembangan produk dapat dimulai dan dijalankan.

### 2.4 Pengkodean Aplikasi

Pada tahap ini, dilakukan pengkodean aplikasi yang berkaitan dengan kebutuhan aplikasi driver pastifresh.id sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya dengan menggunakan bahasa pemrograman java.

### 2.5 Pengujian Program

Pada tahap ini, pengujian dilakukan terhadap kinerja dari aplikasi driver pastifresh.id sejauh mana aplikasi ini dapat digunakan sesuai dengan yang diharapkan, dengan mencoba menjalankan aplikasi, pengujian dilakukan untuk mengetahui kesalahan pada aplikasi dan juga menggunakan kuisioner untuk mengetahui tingkat kepuasan pengguna.

## 3. HASIL DAN PEMBAHASAN

Hasil uji coba aplikasi driver pastifresh.id berbasis Android dilakukan untuk mengetahui efektifitas dan efisiensi aplikasi. Metode pengujian yang digunakan yaitu pengujian program dan membandingkan tingkat akurasi algoritma. Pengujian dilakukan sebanyak 15 kali dengan ketentuan 5 pengujian dengan 2 vendor, 5 pengujian dengan 3 vendor, 5 pengujian dengan 4 vendor..

### 3.1 Pengujian 2 Vendor

Tabel 3.1 merupakan matriks pengujian 1 algoritma floyd warshall dengan 2 vendor.

**Tabel 3.1. Matriks Pengujian 1**

	A	B	C	D
A	0	5,3	7,8	8,7
B	$\infty$	0	2,9	3,4
C	$\infty$	5,1	0	5,4
D	$\infty$	$\infty$	$\infty$	0

Dari pengujian dengan Tabel 3.1 maka kemungkinan menghasilkan Tabel 3.2 berikut ini.

**Tabel 3.2. Kemungkinan Rute Pengujian 1**

No	Rute	Jarak	Total
1.	0-1-2-3	5,3+2,9+5,4	13,6
2.	0-2-1-3	7,8+5,1+3,4	16,3

Rute yang diambil 0-1-2-3 = 5,3+2,9+5,4=13,6

### 3.2 Pengujian 3 Vendor

Tabel 3.3 merupakan matriks pengujian 1 algoritma floyd warshall dengan 3 vendor.

**Tabel 3.3. Matriks Pengujian 1**

	A	B	C	D	E
A	0	7,8	11,5	4,4	5,6000004
B	$\infty$	0	5,6	5,3	5,8
C	$\infty$	4,1	0	7,6	8,2
D	$\infty$	5,0	8,7	0	1,2
E	$\infty$	$\infty$	$\infty$	$\infty$	0

Pada pengujian 3 vendor pada Tabel 3.3 tersebut kemungkinan yang ada dapat dilihat pada Tabel 3.4 berikut ini.

**Tabel 3.4. Kemungkinan Rute Pengujian 1**

No	Rute	Jarak	Total
1.	0-1-2-3-4	7,8+5,6+7,6+1,2	22,2

2.	0-1-3-2-4	7,8+5,3+8,7+8,2	30,0
3.	0-2-1-3-4	11,5+4,1+5,3+1,2	22,100002
4.	0-2-3-1-4	11,5+7,6+5,0+5,8	29,900002
5.	0-3-1-2-4	4,4+5,0+5,6+8,2	23,2
6.	0-3-2-1-4	4,4+8,7+4,1+5,8	23,0

Rute yang diambil 0-2-1-3-4=  $11,5+4,1+5,3+1,2=22,100002$

### 3.3 Pengujian 4 Vendor

Tabel 3.3 merupakan matriks pengujian 1 algoritma floyd warshall dengan 4 vendor.

**Tabel 3.5. Matriks Pengujian 1**

	A	B	C	D	E	F
A	0	4,4	5,3	7,8	11,5	6,4
B	$\infty$	0	2,6	5,0	8,7	2,0
C	$\infty$	3,3	0	2,9	6,6	1,8
D	$\infty$	5,3	5,1	0	5,6	5,0
E	$\infty$	7,6	7,5	4,1	0	7,4
F	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0

Pada pengujian 4 vendor pada Tabel 3.5 kemungkinan yang ada dapat ditunjukkan pada Tabel 3.4 berikut ini.

**Tabel 3.6. Kemungkinan Rute Pengujian 1**

No	Rute	Jarak	Total
1.	0-1-2-3-4-5	$4,4+2,6+2,9+5,6+7,4$	22,9
2.	0-1-2-4-3-5	$4,4+2,6+6,6+4,1+5,0$	22,7
3.	0-1-3-2-4-5	$4,4+5,0+5,1+6,6+7,4$	28,5
4.	0-1-3-4-2-5	$4,4+5,0+5,6+7,5+1,8$	24,3

---

5.	0-1-4-2-3-5	4,4+8,7+7,5+2,9+5,0	28,5
6.	0-1-4-3-2-5	4,4+8,7+4,1+5,1+1,8	24,1
7.	0-2-1-3-4-5	5,3+3,3+5,0+5,6+7,4	26,6
8.	0-2-1-4-3-5	5,3+3,3+8,7+4,1+5,0	26,4
9.	0-2-3-1-4-5	5,3+2,9+5,3+8,7+7,4	29,6
10.	0-2-3-4-1-5	5,3+2,9+5,6+7,6+2,0	23,400002
11.	0-2-4-1-3-5	5,3+6,6+7,6+5,0+5,0	29,5
12.	0-2-4-3-1-5	5,3+6,6+4,1+5,3+2,0	23,3
13.	0-3-1-2-4-5	7,8+5,3+2,6+6,6+7,4	29,7
14.	0-3-1-4-2-5	7,8+5,3+8,7+7,5+1,8	31,099998
15.	0-3-2-1-4-5	7,8+5,1+3,3+8,7+7,4	32,3
16.	0-3-2-4-1-5	7,8+5,1+6,6+7,6+2,0	29,1
17.	0-3-4-1-2-5	7,8+5,6+7,6+2,6+1,8	25,4
18.	0-3-4-2-1-5	7,8+5,6+7,5+3,3+2,0	26,199999
19.	0-4-1-2-3-5	11,5+7,6+2,6+2,9+5,0	29,6
20.	0-4-1-3-2-5	11,5+7,6+5,0+5,1+1,8	31,0
21.	0-4-2-1-3-5	11,5+7,5+3,3+5,0+5,0	32,3
22.	0-4-2-3-1-5	11,5+7,5+2,9+5,3+2,0	29,2

---

23.	0-4-3-1-2-5	11,5+4,1+5,3+2,6+1,8	25,300001
24.	0-4-3-2-1-5	11,5+4,1+5,1+3,3+2,0	26,0

Rute yang diambil 0-1-2-4-3-5=  $4,4+2,6+6,6+4,1+5,0=22,7$

Berdasarkan hasil pengujian yang dilakukan didapat hasil pembuktian algoritma Floyd Warshall pada aplikasi driver pastifresh.id sebagai berikut :

**Tabel. 3.7 Hasil Pengujian**

No	Rute Terpendek	Jarak(Km)	Jalur Terpendek
Pengujian 2 Vendor			
1.	0-1-2-3	13,6	Berhasil
2.	0-1-2-3	8,8	Berhasil
3.	0-1-2-3	14.000001	Berhasil
4.	0-2-1-3	18,1	Berhasil
5.	0-1-2-3	15,5	Berhasil
Pengujian 3 Vendor			
6.	0-2-1-3-4	22,100002	Berhasil
7.	0-3-2-1-4	19,800001	Berhasil
8.	0-1-3-2-4	17,2	Berhasil
9.	0-3-2-1-4	11,799999	Berhasil
10.	0-3-1-2-4	20,4	Berhasil

Pengujian 4 Vendor			
11.	0-1-2-4-3-5	22,7	Berhasil
12.	0-4-2-3-1-5	16,7	Berhasil
13.	0-4-2-3-1-5	16,7	Berhasil
14.	0-12-3-4-5	20,800001	Berhasil
15.	0-1-3-2-4-5	23,9	Berhasil

Algoritma Floyd warshall dapat mengoptimalkan matrik jarak untuk menemukan rute terpendek dari pastifresh ke customer dengan melalui semua titik vendor ditentukan, hal ini berdasarkan hasil pengujian yang menyatakan dan dapat di lihat pada Tabel 3.7.

### 3.4 Prosedur Uji Coba Program

Pada prosedur uji coba program akan dijelaskan tentang cara pengoperasian program serta langkah-langkah yang harus dilakukan untuk menjalankan aplikasi driver pastifresh.id berbasis Android.

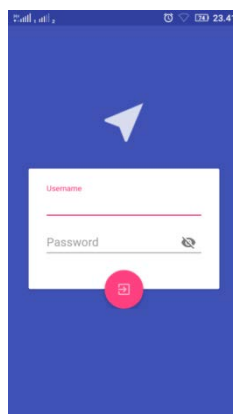
Langkah-langkah yang dilakukan untuk dapat mengoperasikan aplikasi driver pastifresh.id berbasis Android adalah:

1. Meng-install aplikasi driver pastifresh.id berbasis Android, yaitu dengan menginstal driverpastifresh.apk.
2. Setelah aplikasi terinstal pada ponsel Android untuk menjalankan aplikasi , user dapat memulai menggunakan aplikasi pada ponsel.

Langkah awal yang harus dilakukan untuk menjalankan aplikasi driver pastifresh.id berbasis Android ini yaitu dengan membuka aplikasi yang telah diinstal.

1. Menjalankan Halaman Menu Login

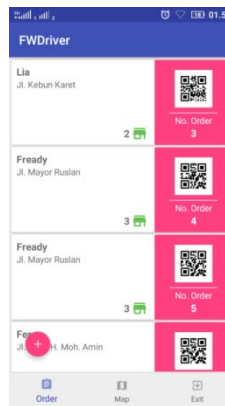
Halaman ini merupakan tampilan menu login. Didalam halaman menu login terdapat *edittext username* dan *password* untuk menginput *username* dan *password* pihak driver. Tampilan *login* dapat dilihat pada Gambar 3.1.



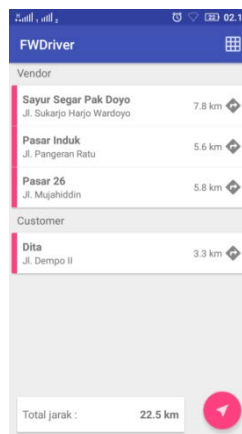


**Gambar 3.1 Login Driver****2. Menjalankan Halaman Data Order**

Tampilan berikut muncul setelah *user* berhasil *login*. Halaman ini merupakan tampilan menu utama sekaligus ketika *fragment order* dipilih. Di dalam halaman ini terdapat seluruh data daftar *order* yang telah masuk ke dalam aplikasi *driver pastifresh.id*. Tampilan Menu Utama atau tampilan ketika menu *fragment order* dipilih dapat dilihat pada Gambar 3.2.

**Gambar 3.2 Data Order****3. Menjalankan Halaman Detail Order**

Tampilan dibawah ini merupakan tampilan ketika user memilih salah satu daftar *order*. Di dalam tampilan ini terdapat *detail order* dan *vendor* mana saja yang harus dikunjungi oleh driver yang telah di urutkan menggunakan algoritma *floyd warshall*. Contoh Tampilan *detail order* dapat dilihat pada Gambar 3.3.

**Gambar 3.3 Detail Order****4. Menjalankan Halaman Membuka Jalan Pada Peta Order**

Tampilan dibawah ini adalah tampilan saat user memilih *button show route in map* yang berada di tampilan *detail order* maka akan tampil sebuah peta dimana

---

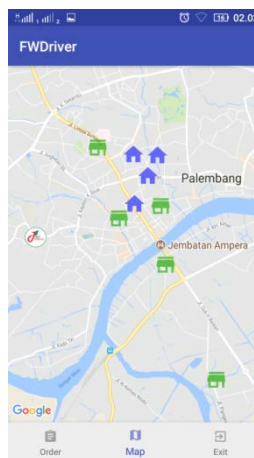
di dalam peta tersebut sudah terbentuk rute yang harus dilewati oleh *driver* dapat dilihat pada Gambar 3.4.



**Gambar 3.4 Membuka Jalan Pada Peta**

5. Menjalankan Halaman Melihat Peta

Halaman berikut muncul saat user memilih fragment Map. Halaman ini menampilkan seluruh vendor yang menjadi mitra pastifresh.id. Rancangan antarmuka Melihat Peta dapat dilihat pada Gambar 3.5.



**Gambar 3.5 Melihat Peta**

#### 4. KESIMPULAN

Berdasarkan hasil implementasi uji coba dan evaluasi dari pembuatan aplikasi ini, dapat dijelaskan kesimpulan dari pengerjaan skripsi ini sebagai berikut :

1. Algoritma Floyd Warshall dapat mengoptimalkan matrik jarak untuk menemukan rute terpendek dari pastifresh ke customer dengan melalui semua titik vendor ditentukan, hal ini dapat dibuktikan keberhasilannya dengan pengujian 2 vendor, 3 vendor dan 4 vendor.
  2. Algoritma ini menjadi algoritma terbaik dalam penentuan jalur saat diimplementasikan dalam aplikasi dengan membantu driver pastifresh.id..
-

## 5. SARAN

Dalam pembuatan aplikasi driver pastifresh.id ini penulis menyadari masih banyak kekurangan yang terdapat dalam aplikasi ini, oleh karena itu penulis berharap aplikasi ini dapat dikembangkan untuk mendapatkan hasil yang lebih baik dari sebelumnya, sehingga penulis menyarankan pengembangan untuk aplikasi ini dapat digabungkan dengan algoritma lain untuk menentukan jalur terbaik dengan parameter yang lebih banyak.

## DAFTAR PUSTAKA

- [1] Ahyar Rivai Hasibuan, 2016, *Penerapan Algoritma Floyd Warshall Dalam Menentukan Rute Terpendek Pada Pemodelan Jaringan Pariwisata di Kota Semarang*, Universitas Negeri Semarang, Indonesia.
  - [2] Arsyad, Azhar M.A 2011, *Media Pembelajaran*, Rajawali Pers, Jakarta.
  - [3] A.S Rosa dan M. Shalahuddin 2013, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Informatika, Bandung.
  - [4] Dharma Kasman, Akhmad 2013, *Kolaborasi Dahsyat Android dengan PHP dan MYSQL*, Lokomedia, Yogyakarta.
  - [5] Friska Widya Ningrum & Tatyantoro Andrasto, 2016, *Penentuan Jarak Terpendek Rute Tranmusi Dengan Algoritma Floyd Warshall*, SEMANTIK, Palembang.
  - [6] Komputer, Wahana 2013, *Android Programming With Eclipse*, Andi Offset, Yogyakarta.
  - [7] Muraya, Yosep 2013, *Pemrograman Android Black Box*, Jasakom, Jakarta.
  - [8] M. Zani Wibowo & Lutfi Pratama, 2016, *Rancang Bangun Aplikasi Permainan “Help Your Mom “ Menggunakan Algoritma Floyd Warsall*, STMIK GI MDP, Palembang.
  - [9] M. Yudi Kriswanto, R. Kristoforus Jawa Bendi & Arif Aliyanto, 2014, *Penentuan Jarak Terpendek Rute Tranmusi Dengan Algoritma Floyd Warshall*, SEMANTIK, Palembang.
  - [10] Pressman Ph.D, Roger S 2012, *Rekayasa Perangkat Lunak*, Andi Offset, Yogyakarta.
  - [11] Simarmata, Janner 2010, *Rekayasa Perangkat Lunak*, Andi Offset, Yogyakarta.
  - [12] Suprianto, Dodi 2012, *Pemrograman Aplikasi Android*, Mediakom, Yogyakarta.
-