

IMPLEMENTASI LOCATION BASED SERVICE DAN ALGORITMA POLYLINE PADA APLIKASI PEMESANAN MOBIL DI UNIVERSITAS BUDI LUHUR BERBASIS ANDROID

Yusup Supriyadi¹⁾, Painem²⁾

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

E-mail : yusupubl14@gmail.com¹⁾, painem@budiluhur.ac.id²⁾

Abstrak

Universitas Budi Luhur adalah sebuah universitas swasta di Jakarta yang berdiri sejak 1 April 1979. Di Universitas Budi Luhur beberapa dosen sering menggunakan fasilitas transportasi yang disediakan oleh pihak kampus. Namun di era teknologi informasi dan komunikasi yang semakin maju, dosen yang ingin menggunakan fasilitas transportasi yang ada masih harus menggunakan cara manual dengan memesan melalui staff yang bertugas dibagian transportasi. Untuk mengatasi masalah tersebut maka dibuatkanlah aplikasi yang dapat mempermudah proses pemesanan mobil bagi dosen. Selain untuk dosen aplikasi ini juga dapat digunakan oleh staff transportasi dalam hal ini bisa disebut sebagai bos, bos dapat melihat daftar pesanan masuk dan bos akan mengirimkan pesan yang masuk kepada driver untuk diproses. Pada aplikasi ini akan ditampilkan lokasi pemesanan dan lokasi tujuan dari pesan sehingga akan memudahkan driver dalam melakukan proses penjemputan dan pengantaran. Semua proses mulai dari pemesanan mobil sampai konfirmasi pesanan selesai diantar oleh driver dilakukan dalam 1 (satu) aplikasi. Aplikasi ini sendiri merupakan aplikasi berbasis android yang menggunakan teknologi Location Based Service dan GPS (Global Positioning System) untuk menentukan lokasi penjemputan user serta menampilkan rute menggunakan algoritma polyline, untuk memudahkan driver dalam melakukan proses penjemputan dan pengantaran ke lokasi tujuan.

Kata kunci: Location Based Service, GPS, android, polyline.

1. PENDAHULUAN

Teknologi komunikasi dan informasi mengalami perkembangan yang sangat pesat terutama yang berbasis android dimana setiap orang saat ini hampir semuanya memiliki *smartphone*, terbukti dari banyaknya aplikasi yang muncul untuk membantu para pengguna dalam berbagai hal. Namun di Universitas Budi Luhur proses pemesanan mobil masih dilakukan dengan cara lama yaitu datang langsung atau menghubungi bagian transportasi, belum ada media yang dapat membuat proses pemesanan menjadi lebih mudah dan efisien.

Oleh karena itu dikembangkan aplikasi *mobile* yang dapat melakukan 3 proses dalam 1 aplikasi yaitu proses pemesanan mobil oleh pemesan, proses penerimaan pesan masuk dan meneruskan pesan tersebut ke *driver* dilakukan oleh staff transportasi dalam hal ini bisa disebut sebagai bos, kemudian *driver* dapat menerima pesan masuk yang telah dikirimkan oleh bos, kemudian *driver* bertugas menyelesaikan pesan yang telah diberikan.

Aplikasi ini mengimplementasikan teknologi *Location Based Service* dan GPS (*Global Positioning System*) untuk menampilkan lokasi pengguna dan mempermudah dalam pencarian lokasi untuk lokasi penjemputan dan lokasi tujuan.

Untuk driver akan ditampilkan rute pada aplikasi dengan menggunakan algoritma *polyline* untuk memudahkan proses penjemputan dan pengantaran ke lokasi tujuan, selain rute akan ditampilkan juga jarak dan waktu tempuhnya.

2. METODE PENELITIAN

2.1. Metode Penelitian

Berikut ini adalah rincian tahapan dalam pembuatan aplikasi pemesanan mobil berbasis Android yaitu:

- Pengumpulan Data
Mengumpulkan kebutuhan dari keseluruhan elemen sistem yang akan diaplikasikan ke dalam bentuk software atau perangkat lunak.
- Menganalisa Kebutuhan Aplikasi
Setelah memperoleh kebutuhan aplikasi kemudian dipelajari dan dianalisa mengenai fungsi-fungsi apa saja yang diperlukan untuk mengimplementasikan aplikasi ini.
- Desain atau Perancangan Aplikasi
Merancang tampilan aplikasi yang akan dibangun sesuai dengan kebutuhan aplikasi sehingga dapat mempermudah dalam proses pengkodean.
- Pengkodean
Pengkodean dilakukan untuk memudahkan dalam mengimplementasikan rancangan aplikasi dengan menggunakan bahasa pemrograman Java di Android Studio.
- Implementasi
Rancangan aplikasi yang sudah dibuat kemudian diimplementasikan berdasarkan analisa masalah.
- Pengujian
Pengujian dilakukan setelah aplikasi selesai dibuat dengan melakukan beberapa pengujian program dan mencari kesalahan pada program ini

hingga tidak ada lagi kesalahan program dan program sudah berjalan sesuai dengan yang dirancang.

Aplikasi *mobile* yang dikembangkan ini berbasis Android. Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi [1]. Pengiriman dan pengambilan data dilakukan dengan menggunakan *library Retrofit* yang tersedia pada Android. *Resource* yang berbentuk *JSON* ini disediakan oleh *REST API server* yang mana dapat dimanfaatkan oleh aplikasi Android dengan *library Retrofit*. *JSON (Java Script Object Notation)* adalah format pertukaran data yang bersifat ringan [2].

Aplikasi pemesanan mobil yang dikembangkan ini mengimplementasikan teknologi *Location Based Service* dan *Global Positioning System* untuk menampilkan lokasi pengguna serta pencarian lokasi penjemputan dan lokasi tujuan. *Location Based Service (LBS)* atau layanan berbasis lokasi adalah layanan informasi yang dapat diakses menggunakan piranti mobile melalui jaringan internet dan selular serta memanfaatkan kemampuan penunjuk lokasi pada piranti mobile [3]. *LBS* terdiri dari 5 komponen utama diantaranya *mobile devices, communication network, position component, Service and Application Provider, data and content provider* [4]. Lokasi dalam *LBS* dapat dinyatakan dalam bentuk data spasial atau deskripsi teks [5]. Data spasial dapat dinyatakan menggunakan latitude, longitude dan altitude [5]. *GPS (Global Positioning System)* adalah sistem navigasi untuk penentuan posisi dengan menggunakan satelit.

Algoritma *Polyline* digunakan dalam mengembangkan aplikasi ini untuk menampilkan rute sehingga memudahkan *driver* dalam melakukan proses penjemputan dan pengantaran. Algoritma *polyline* adalah algoritma yang digunakan untuk menampilkan atau menunjukkan rute tertentu antara dua titik pada map Google [6]. *Encode polyline* adalah satu set pasangan koordinat yang telah diubah menjadi *string ASCII* untuk secara signifikan mengurangi keseluruhan ukuran data. *Polyline* yang dikodekan digunakan untuk menyimpan kumpulan data koordinat besar untuk memproyeksikan garis atau bentuk pada peta. Koordinat titik diencodekan menggunakan nilai yang dimasukan. Format *encode* di dalam *polyline* harus mewakili dua koordinat yang merepresentasikan garis lintang dan bujur hingga presisi yang wajar. Langkah-langkah mengencode nilai pada algoritma *polyline* :

- a. Ambil nilai awal misal:
-179.9832104
- b. Ambil nilai desimal dan kalikan dengan 1e5 dan bulatkan hasilnya:
-17998321
- c. Konversikan nilai desimal menjadi biner 32bit. Perhatikan bahwa nilai negatif harus dihitung menggunakan pelengkap keduanya dengan

membalik nilai biner dan menambahkan satu ke hasilnya:

```
00000001 00010010 10100001 11110001
11111110 11101101 01011110 00001110
11111110 11101101 01011110 00001111
```

- d. Geser nilai biner ke kiri sebanyak satu bit:
11111101 11011010 10111100 00011110
- e. Jika nilai desimal aslinya negatif, balik encode berikut:
00000010 00100101 01000011 11100001
- f. Pisahkan nilai biner menjadi potongan 5-bit (mulai dari sisi sebelah kanan):
00001 00010 01010 10000 11111 00001
- g. Tempatkan potongan 5-bit dalam urutan mundur:
00001 11111 10000 01010 00010 00001
- h. OR setiap nilai dengan 0x20 :
100001 111111 110000 101010 100010 000001
- i. Konversikan setiap nilai ke desimal:
33 63 48 42 34 1
- j. Tambahkan 63 ke setiap nilai:
96 126 111 105 97 64
- k. Konversikan setiap nilai ASCII:
`~oia@

Table 1. Tabel ASCII

Dec	Symbol	Dec	Symbol	Dec	Symbol	Dec	Symbol
0	NULL	32	(Space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(72	H	104	h
9	HT	41)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

Aplikasi ini menggunakan API Google Map V2 untuk menampilkan maps. Mulai tahun 2013, aplikasi Android yang ingin menampilkan Google Map harus menggunakan layanan Google Map V2. Ada perbedaan yang mencolok dalam penerapan source code antara Google Map Android V1 dengan Google Map Android V2. Mulai dari penggunaan SHA1 yang menggantikan MD5 untuk mendapatkan Google API key hingga penggunaan *Fragment* yang menggantikan *MapView*. *Developer* juga harus menginstall library *google-play-service* terlebih dahulu pada Android SDK.

Pendeteksian lokasi pengguna juga dapat ditambahkan pada pengembangan aplikasi yang menggunakan Google Map, baik menggunakan GPS, jaringan seluler, maupun jaringan wifi.

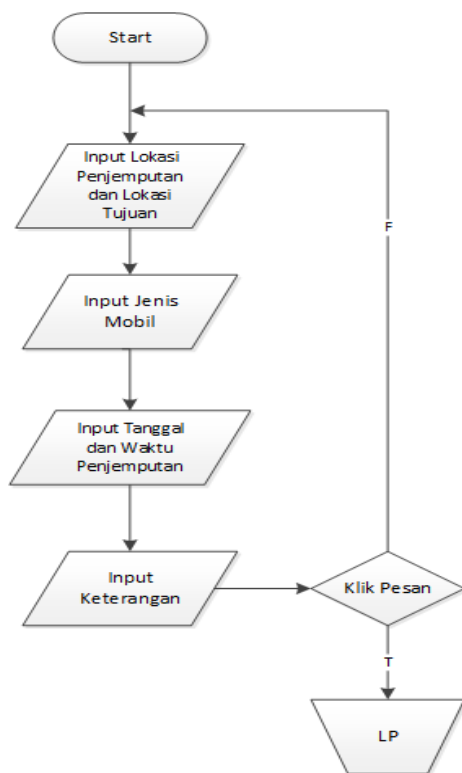
Google Play Service adalah sebuah komponen yang terinstal *default* di semua perangkat berbasis Android *System*. Dalam mengembangkan aplikasi ini *Google Play Service* harus diinstall terlebih dahulu untuk dapat memunculkan peta google pada aplikasi.

Google USB Driver digunakan untuk pengujian langsung di *device* Android melalui kabel data pada saat pengembangan aplikasi Android.

Google Play Service dan *Google USB driver* adalah 2 (dua) library yang dibutuhkan dalam pengembangan aplikasi Android *Location Based Service* yang menampilkan peta.

2.2. Flowchart dan Algoritma Program

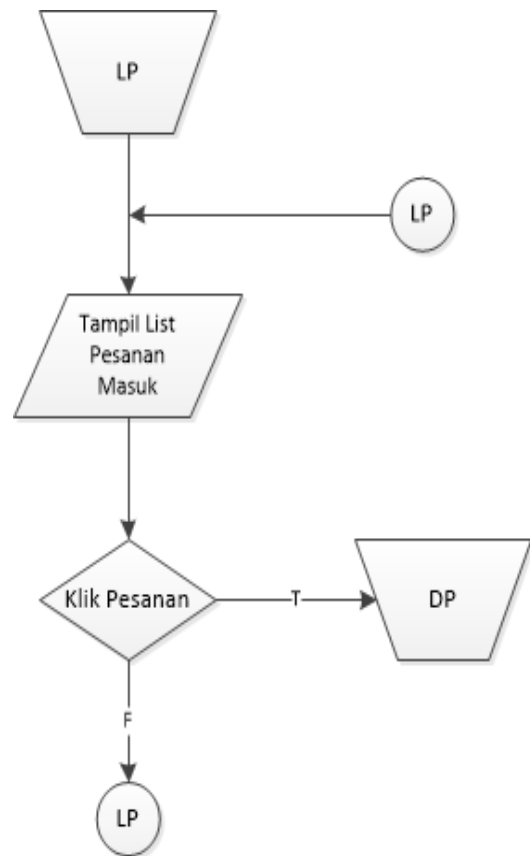
Berikut adalah *flowchart* mulai dari proses pemesanan sampai berhasil diantarkan oleh driver :



Gambar 1. Flowchart Proses Pemesanan

Algoritma proses pemesanan :

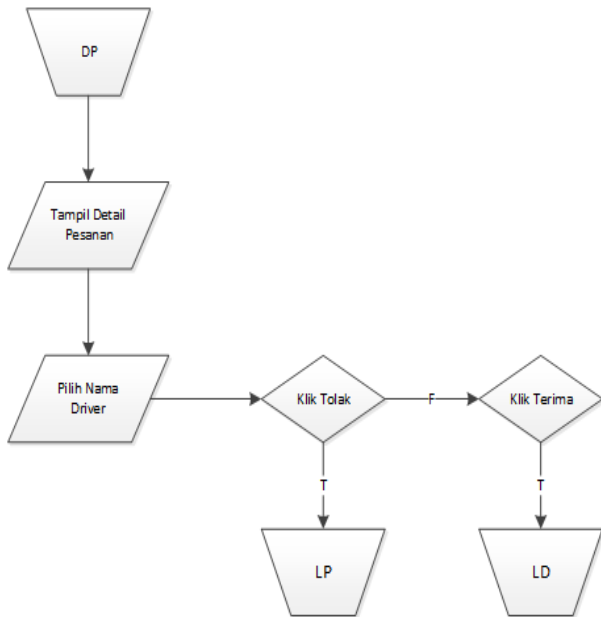
1. Tampil halaman Pemesanan
2. *Input* Lokasi Penjemputan
3. *Input* Lokasi Tujuan
4. *Input* Jenis Mobil
5. *Input* Tanggal Penjemputan
6. *Input* Waktu Penjemputan
7. *Input* Keterangan
8. Klik Pesan
9. *If* Semua sudah diinput *Then*
10. Kirim Pesanan
11. *Else*
12. Lengkapi *form* pemesanan
13. *End if*



Gambar 2. Flowchart List Pesanan

Algoritma list pesanan masuk :

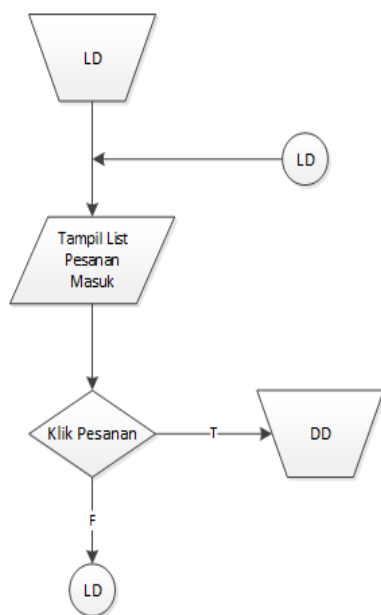
1. Tampil list pesanan
2. Pilih *action*
3. *If action* = klik salah satu pesanan *Then*
4. Tampil detail pesanan
5. *Else*
6. Lihat list pesanan
7. *End if*



Gambar 3. Flowchart Detail Pesanan

Algoritma detail pesanan masuk :

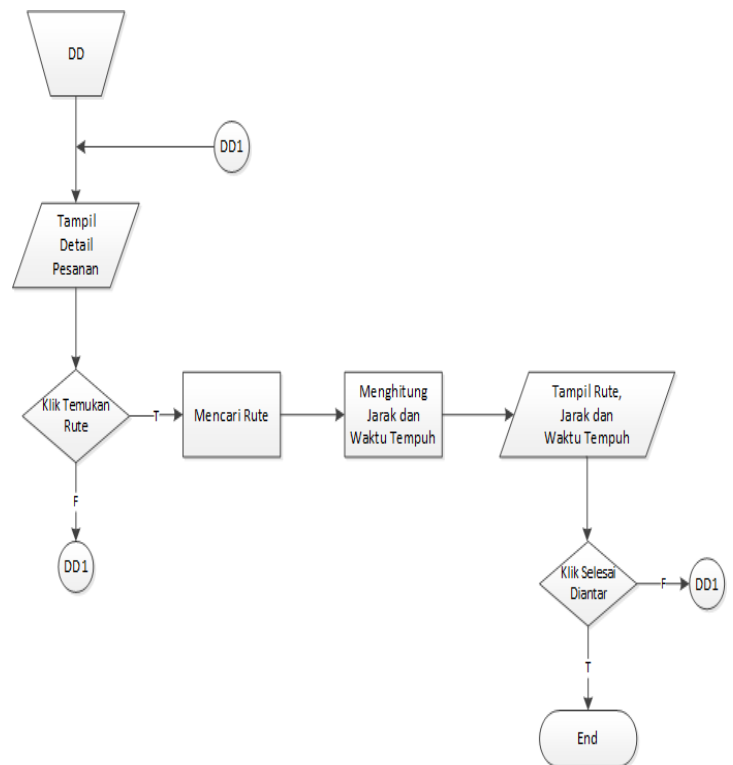
1. Tampil halaman Detail Pesanan
2. Lihat data lengkap pesanan
3. Pilih action
4. *If action = klik button Terima Then*
5. Input nama driver yang akan ditugaskan
6. Kirim pesan ke driver yang ditugaskan
7. *Else if action = klik button Tolak Then*
8. Tolak pesan kirim notifikasi ke pemesan
9. *Else*
10. Kembali ke list pesanan masuk
11. *End if*



Gambar 4. Flowchart List Pesanan Untuk Driver

Algoritma list pesanan masuk untuk driver :

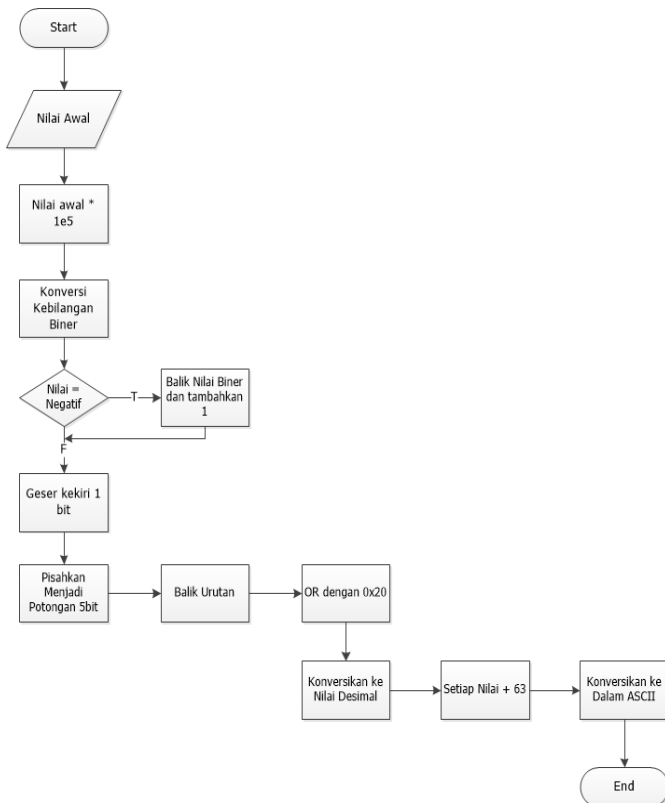
1. Tampil list pesanan
2. Pilih action
3. *If action = klik salah satu pesanan Then*
4. Tampil detail pesanan
5. *Else*
6. Lihat list pesanan
7. *End if*



Gambar 5. Flowchart Detail Pesanan Untuk Driver

Algoritma detail pesanan masuk untuk driver :

1. Tampil halaman Detail Pesanan
2. Lihat data lengkap pesanan
3. Pilih action
4. *If action = klik Temukan Rute Then*
5. Output rute pada map, jarak tempuh, waktu tempuh
6. *if action = klik Marker Jemput Then*
7. Direct google map tampil rute ke lokasi penjemputan
8. *Else*
9. Direct ke google map tampil rute ke lokasi tujuan
10. *Else if action = klik Selesai Diantar Then*
11. Konfirmasi Pesanan ke bos
12. *Else*
13. Kembali ke list pesanan masuk
14. *End if*



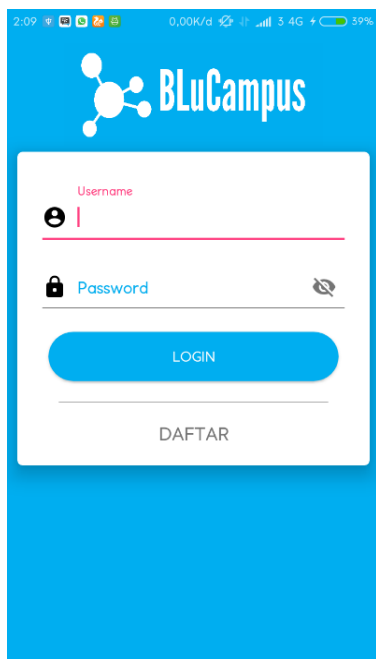
Gambar 6. Flowchart Algoritma Polyline

3. HASIL DAN PEMBAHASAN

3.1. Tampilan Layar

Disini akan dijelaskan mengenai tampilan-tampilan layar dari awal proses pemesanan sampai pesanan berhasil diantarkan oleh *driver* :

3.1.1 Tampilan Layar Form Login



Gambar 7. Tampilan Layar Form Login

Gambar 7 merupakan tampilan layar *form login* dimana pengguna harus melakukan *login* terlebih dahulu dengan cara memasukkan *username* dan *password* yang sesuai.

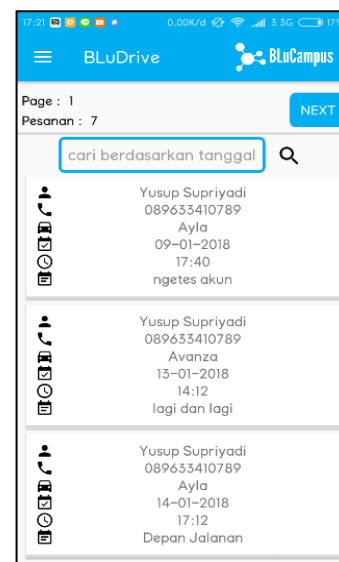
3.1.2 Tampilan Layar Form Pemesanan



Gambar 8. Tampilan Layar Form Pemesanan

Gambar 8 merupakan tampilan layar *form pemesanan*, pemesan diharuskan memasukkan lokasi penjemputan, lokasi tujuan, jenis mobil, tanggal penjemputan, waktu penjemputan, dan keterangan untuk si *driver* lalu klik pesan untuk mengirimkan pesanan.

3.1.3 Tampilan Layar List Pesanan



Gambar 9. Tampilan Layar List Pesanan

Gambar 9 merupakan tampilan layar *list pesanan*, pada tampilan ini bos dapat melihat daftar pesanan yang masuk.

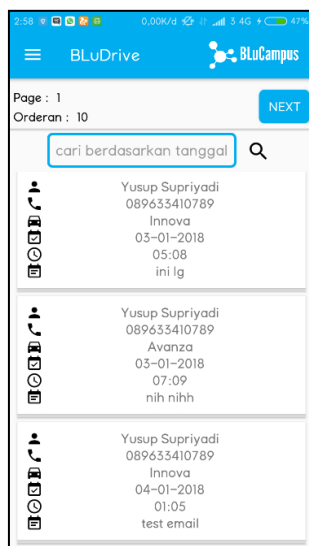
3.1.4 Tampilan Layar Form Detail Pesanan



Gambar 10. Tampilan Layar Form Detail Pesanan

Gambar 10 merupakan tampilan layar *form* Detail pesanan, untuk melihat detail pesanan bos harus mengklik salah satu pesanan yang ada di *list* pesanan masuk. Pada *form* detail pesanan bos dapat melihat informasi pesanan secara detail untuk menerima dan meneruskan pesanan ke *driver* bos harus memilih nama *driver* terlebih dahulu setelah itu klik Terima makan pesanan akan dikirimkan ke *driver* yang di tuju. Untuk menolak pesanan bos tinggal mengklik Tolak maka akan mengirimkan notifikasi berupa email kepada pemesan bahwa pesanan tidak dapat diproses.

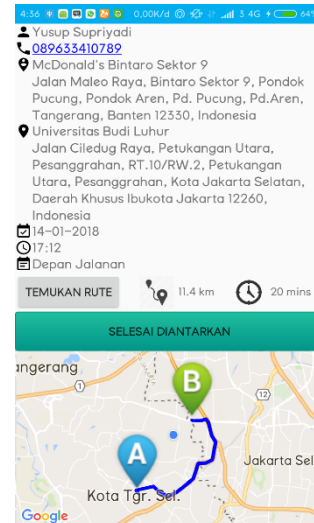
3.1.5 Tampilan Layar List Pesanan Driver



Gambar 11. Tampilan Layar List Pesanan

Gambar 11 merupakan tampilan layar *list* pesanan untuk *driver*, pada tampilan ini *driver* dapat melihat daftar pesanan masuk yang dikirimkan oleh bos.

3.1.6 Tampilan Layar Form Detail Pesanan Driver



Gambar 12. Tampilan Layar Form Detail Pesanan Driver

Gambar 12 merupakan tampilan layar *form* Detail pesanan untuk *driver*, untuk melihat detail pesanan *driver* harus mengklik salah satu pesanan yang ada di *list* pesanan. Pada *form* detail pesanan *driver* dapat melihat informasi pesanan secara detail. Klik Temukan Rute untuk melihat rute, jarak tempuh, dan waktu tempuh dari lokasi penjemputan ke lokasi tujuan. Setelah pemesan berhasil diantarkan ke lokasi tujuan *driver* harus mengklik *button* Selesai Diantarkan untuk konfirmasi bahwa pesanan sudah selesai dikerjakan.

3.2 Tanggapan Pengguna

Dari beberapa responden yang mencoba menggunakan aplikasi ini dan memberikan tanggapannya di dalam kuesioner yang telah penulis buat. Penulis membagi beberapa pertanyaan penilaian menjadi 4 kelompok, yaitu *functionality* (kegunaan), *reliability* (kehandalan), *usability* (kemudahan penggunaan), dan *efficiency* (efisiensi). Berikut adalah hasil dari kuesioner yang telah diisi oleh 20 orang responden.

Dari 4 kategori aspek penilaian skor dari 20 responden dapat dilihat pada Tabel 2 sebesar 95 jawaban sangat setuju, 207 jawaban setuju, 18 jawaban ragu-ragu, 0 untuk jawaban tidak setuju dan sangat tidak setuju.

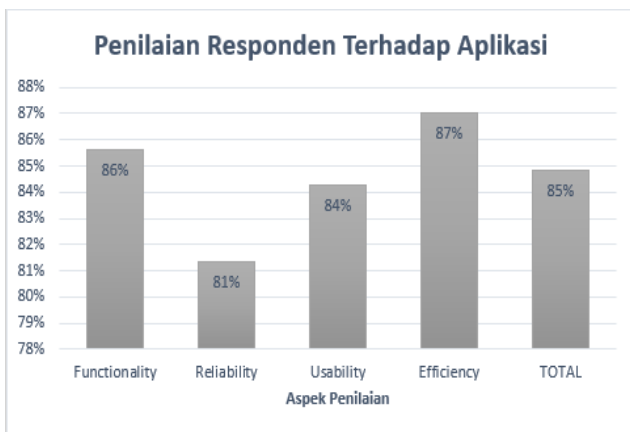
Table 2. Tabel Skor Responden

Aspek Penilaian	Skor Responden				
	5	4	3	2	1
	SS	S	R	TS	STS
Functionality	30	68	2	-	-
Reliability	14	36	10	-	-
Usability	22	53	5	-	-
Efficiency	29	50	1	-	-
Jumlah	95	207	18	-	-

Pada Tabel 3 hasil penilaian responden terhadap aplikasi dari 4 kategori aspek penilaian total skor aktual untuk aspek *functionality* sebesar 86% ,aspek *reliability* 81%, aspek *usability* 84% dan aspek *efficiency* 87%, dan total skor aktual keseluruhan sebesar 85%.

Table 3. Tabel Skor Aktual

Aspek Penilaian	Skor Aktual					Total Skor Aktual	Skor Ideal	%
	5	4	3	2	1			
	SS	S	R	TS	STS			
Functionality	150	272	6	-	-	428	500	86%
Reliability	70	144	30	-	-	244	300	81%
Usability	110	212	15	-	-	337	400	84%
Efficiency	145	200	3	-	-	348	400	87%
Total	475	828	54	-	-	1357	1600	85%



Gambar 13. Diagram Penilaian Responden

Gambar 13 menunjukkan diagram penilaian responden aplikasi dari 4 aspek penilaian dan total keseluruhan.

3.3 Evaluasi Program

Kelebihan Program :

- Aplikasi ini berbasis Android yang bisa di install di *smartphone* sehingga lebih praktis dalam proses pemesanan mobil.
- Memudahkan *driver* dalam mencari lokasi pemesan.
- Memiliki tampilan yang mudah dipahami.
- Aplikasi mudah dimengerti oleh pengguna.
- Data yang terkirim pada saat pemesanan sampai di terima oleh *driver* masih akurat atau sama.

Kekurangan Program

- Beberapa *smartphone* mengalami *crash* saat menjalani aplikasi ini.
- Semua proses yang dilakukan bergantung pada kondisi internet yang digunakan.
- Pada beberapa kondisi rute terkadang tidak tampil.

4. KESIMPULAN

Dengan mengimplementasikan teknologi *Location Based Service* dan GPS (*Global Positioning System*) pada aplikasi ini maka memudahkan dalam mencari lokasi untuk penjemputan dan juga lokasi tujuan pada saat proses pemesanan oleh pemesan.

Algoritma *polyline* yang digunakan membantu *driver* untuk melihat rute, jarak tempuh, dan juga waktu yang tempuh dari lokasi penjemputan ke lokasi tujuan.

Pada aplikasi ini semua proses mulai dari pemesanan hingga pesanan berhasil diantarkan dilakukan dalam 1 aplikasi yang dapat di install di *smartphone* pengguna sehingga proses pemesanan mobil menjadi lebih mudah dan efisien.

5. DAFTAR PUSTAKA

- [1] Safaat, N., 2015. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. 2 ed. Bandung: Penerbit Informatika.
- [2] Bhakti Destian Wijaya, F. E. M. A. A. F., 2015. IMPLEMENTASI JSON PARSING PADA APLIKASI MOBILE E-COMMERCE Studi Kasus : CV V3 Tekno Indonesia. *Pseudocode*, 2(1), pp. 1-9.
- [3] Almuzakki, M. A., 2013. Rancang Bangun Aplikasi Location Based Service Pencarian Lokasi Wisata Di Kota Semarang Berbasis Android. Dokumen karya ilmiah, pp. 1-8.
- [4] Putra, A. N., Tambunan, T. D. dan Ramadhan, K. N., 2011. Aplikasi Wisata Kota Bandung Menggunakan Metode Location Based Services (LBS) pada Android, Politeknik Telkom.
- [5] Anwar, S. N., Nugroho, I. dan Lestariningsih, D. E., 2015. Perancangan Dan Implementasi Aplikasi Mobile Semarang Guidance Pada Android, *Jurnal Teknologi Informasi DINAMIK*, 20(201), hal. 148-158.
- [6] Vishal Bharte, K. P. L. J. D. J., 2014. Bus Monitoring System Using Polyline Algorithm. *Internasional Journal of Scientific and Research Publications*, 4(4), pp. 1-5.