

**SISTEM INFORMASI JASA *CUSTOMER SERVICE* PADA  
SHOES AND CARE LAMPUNG BERBASIS ANDROID**

**Skripsi**

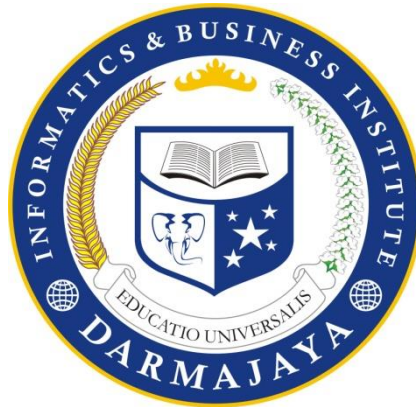
Diajukan Sebagai Salah Satu Syarat Untuk Mencapai Gelar  
SARJANA KOMPUTER  
Pada Program Studi Sistem Informasi  
INFORMATICS INSTITUTE AND BUSINESS DARMAJAYA



Oleh

**Muamar Nabila**  
**1511050011**

**JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
INFORMATICS INSTITUTE AND BUSINESS DARMAJAYA  
BANDAR LAMPUNG  
2019**



### **PERNYATAAN**

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggungjawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 24 Oktober 2019

**Muamar Nabila**  
**NPM. 1511050011**

## HALAMAN PERSETUJUAN

Judul Skripsi : SISTEM INFORMASI JASA *CUSTOMER SERVICE* PADA SHOES AND CARE LAMPUNG BERBASIS ANDROID

Nama Mahasiswa : Muamar Nabila  
NPM : 1511050011  
Program Studi : S1 Sistem Informasi

Disetujui Oleh :

Dosen Pembimbing

Ketua Program Studi  
Sistem Informasi

**hi Marshella F, S.Kom., M.T.I**  
NIK. 11840510

**Nurjoko, S.Kom., M.T.I**  
NIK. 00440702

**HALAMAN PENGESAHAN**

Telah Diuji dan Dipertahankan Didepan Tim Penguji Skripsi  
Program Studi Sistem Informatics Institute And Business Darmajaya  
Bandar Lampung dan Dinyatakan Diterima untuk Memenuhi Syarat Guna  
Memperoleh Gelar Sarjana Komputer

**MENGESAHKAN**

- |                                  |              |
|----------------------------------|--------------|
| 1. Tim Penguji                   | Tanda Tangan |
| Penguji 1 : Indera, S.Kom.,M.T.I | .....        |
| Penguji 2 : TM. Zaini, M.Kom     | .....        |
| 2. Dekan Fakultas Ilmu Komputer  |              |

**Srivanto, S.Kom., MM., Ph.D**  
**NIK. 00210800**

Tanggal Lulus Ujian : 20 September 2019

## HALAMAN PERSEMBAHAN

Segala yang kuraih adalah kehendak Allah SWT dan bukti kasih sayang dari orang-orang yang menyayangiku, dengan mengucapkan syukur kepada Allah SWT dan atas segala limpahan nikmat Allah SWT kepadaku dan segala ketulusan dan kerendahan hati kupersembahkan sebuah karya kecil hasil jerih payah perjuanganku ini untuk :

- Kedua Orangtuaku tercinta yakni, Ayahanda Krinsa Bahar dan Ibunda Ponirah, yang tiada pernah hentinya selama ini memberiku do'a, dorongan, kasih sayang, nasehat, dana, bimbingan, semangat, serta ketabahan yang telah diberikan dan kasih sayang yang begitu besar yang tak dapat terbalaskan walau sampai akhir hayatku. Semoga sehat selalu dan panjang umur orangtuaku.
- Kepada seluruh keluargaku yang telah mendukung, terima kasih khususnya kepada kakakku Alm.Ariyo Prassetiyo, Himawan Sutanto, adik- adikku M Rizki Ramadhan, Naswa Adinda dan M Arif Noval tercinta atas doa, semangat dan dukungannya.
- Teman–teman seperjuangan dan sahabat–sahabatku terima kasih telah memberikan bantuan, hiburan, motivasi, kritik dan saran yang positif. Untuk semua mahasiswa/i SI angkatan 2015.
- Almamaterku tercinta Informatics Institute and Bussiness Darmajaya Bandar Lampung.

**MOTTO**

*Jika Orang lain bisa, maka aku juga termasuk Bisa.*

*Selalu Berjuang, Ikhlas, Sabar, dan Bersyukur.*

*(Muamar Nabila)*

## INTISARI

### SISTEM INFORMASI JASA *CUSTOMER SERVICE* PADA SHOES AND CARE LAMPUNG BERBASIS ANDROID

Oleh:

**MUAMAR NABILA**

e-mail : [muamarnabila@gmail.com](mailto:muamarnabila@gmail.com)

*Sistem informasi dan teknologi komputer berkembang sangat pesat. Sejalan dengan besarnya kebutuhan terhadap informasi. Perkembangan teknologi informasi tidak lepas dari pesatnya perkembangan teknologi komputer. Karena komputer merupakan media yang dapat memberikan kemudahan bagi manusia dalam menyelesaikan pekerjaannya. Berdasarkan informasi yang didapat permasalahan pada Shoes and Care Lampung, Shoeshine dan Dropit Clean menunjukkan bahwa dalam proses bisnis yang berjalan saat ini masih memiliki kekurangan dikarenakan pengelola toko kesulitan dalam menyebarkan informasi serta menampilkan lokasi tempat usaha dan kurangnya efisien waktu pada pelanggan dalam penggunaan layanan jasa.*

*Berdasarkan permasalahan diatas tujuan dari penelitian ini adalah untuk membantu pihak pengelola Toko dalam membangun sebuah sistem informasi pemesanan jasa lewat android serta bisa menjadi media penyebaran informasi layanan, lokasi yang mudah didapatkan dan meningkatkan kinerja mutu layanan jasa yang disediakan oleh Shoes and Care Lampung, Shoeshine dan Dropit Clean. Oleh karena itu metode yang akan dipakai ialah menggunakan metode pengumpulan data seperti observasi, wawancara, dan studi pustaka serta metode pengembangan sistem menggunakan Waterfall dengan 4 tahapan yang digunakan yaitu requirement system, system design, implementation, dan integration & testing.*

*Hasil dari penelitian ini adalah membantu pihak pengelola Toko Shoes and Care Lampung, Shoeshine, Dropit Clean dalam mengembangkan sistem informasi pemesanan jasa , menampilkan lokasi toko dan penyebaran promosi.*

***Kata Kunci : Sistem informasi, Shoes and Care, Android.***

## ABSTRACT

### ANDROID BASED CUSTOMER SERVICE INFORMATION SYSTEMS IN SHOES AND CARE LAMPUNG

By:

**MUAMAR NABILA**

e-mail: muamarnabila@gmail.com

Information systems and computer technology are developing very rapidly. In line with the large need for information, the development of information technology can not be separated from the rapid development of computer technology because computers are media that can provide convenience for humans in completing their work. Based on the information obtained by the problem in Lampung Shoes and Care, Shoeshine and Drop it Clean shows that the current business process still has shortcomings due to the difficulty of store managers in disseminating information and displaying business locations and the lack of time efficiency for customers in using services.

Based on the above problems the objectives of this study were to assist the store manager in building a service information system, easily accessible location and improve the quality of service performance provided by Shoes and Care Lampung, Shoeshine and Drop it Clean. Therefore, the method used was to use the data collection methods through observation, interviews and literature studies as well as the systems development methods used *waterfalls* with 4 stages used, namely system requirements, system design, implementation, and integration and testing.

The results of this study can help the managers of Lampung Shoe and Care stores, Shoeshine and Drop it Clean in developing the service ordering information system, displaying store locations and distributing promotions.

**Keywords: Information Systems, Shoes and Care, Android**



## **RIWAYAT HIDUP**

### **1. Identitas**

- a. Nama : Muamar Nabila
- b. NPM : 1511050011
- c. Tempat/Tanggal Lahir : Bandar Lampung, 10 Juli 1997
- d. Agama : Islam
- e. Alamat : Jl. Maulana Yusuf No.38,  
Tanjung Karang Pusat, Bandar Lampung
- f. Suku : Lampung
- g. Kewarganegaraan : Indonesia
- h. E-Mail : muamarnabila@gmail.com
- i. HP : 085758221228

### **2. Riwayat Pendidikan**

- a. TK : TK Persit Kartika II-5 Bandar Lampung
- b. Sekolah Dasar : SD Kartika II-5 Bandar Lampung
- c. Sekolah Menengah Pertama : SMP PGRI 1 Bandar Lampung
- d. Sekolah Menengah Atas : SMK Negeri 4 Bandar Lampung
- e. Sarjana : Informatics Institute and Bussiness  
Darmajaya Jurusan Sistem Informasi  
Fakultas Ilmu Komputer

Dengan ini saya menyatakan bahwa semua keterangan yang saya sampaikan diatas adalah benar.

Bandar Lampung, 24 Oktober 2019  
Yang Menyatakan,

**Muamar Nabila**  
**NPM. 1511050011**

## PRAKATA

Puji syukur kehadirat ALLAH SWT, yang telah memberikan rahmat dan hidayah-Nya. Sehingga penulis dapat menyelesaikan Skripsi yang disusun sebagai syarat untuk mencapai gelar sarjana strata 1 pada jurusan Sistem Informasi Informatics Institute and Business Darmajaya Bandar Lampung. Dalam penulisan skripsi ini penulis banyak memperoleh bimbingan dan bantuan dari berbagai pihak. Untuk itu penulis mengucapkan terimakasih kepada:

1. ALLAH SWT tiada kata terucap, selain memuji keagungan-Mu dan mengucap Alhamdulillah atas segala Nikmat, Rahmat dan Hidayah yang telah Engkau berikan.
2. Nabi Muhammad SAW yang telah membawa manusia dari alam yang penuh dengan kekufuran ke alam yang terang benderang.
3. Ayah dan Ibu tercinta, terima kasih atas do'a, kasih sayang, nasehat, dukungan moril maupun materi, bimbingan dan semangat serta ketabahan yang telah diberikan untuk anak-anaknya agar menjadi orang yang berguna baik dunia maupun akherat.
4. Bapak Ir. Firmansyah Y. Alfian, MBA., MSc Selaku Rektor Informatics Institute and Business Darmajaya.
5. Bapak Sriyanto, S.Kom.,MM,. Ph.D Selaku Dekan Fakultas Ilmu Komputer Informatics Institute and Business Darmajaya.
6. Bapak Nurjoko, S.Kom., M.T.I. Selaku Ketua Jurusan Sistem Informasi.
7. Ibu Ochi Marshella F, S.Kom., M.T.I selaku pembimbing utama yang telah banyak meluangkan waktunya dalam memberikan bimbingan dan

pengarahan kepada saya sehingga skripsi ini dapat diselesaikan dengan baik.

8. Bapak Indera, S.Kom.,M.T.I dan Bapak TM. Zaini, M.Kom selaku dosen penguji skripsi.
9. Semua dosen Program Studi Sistem Informasi terima kasih telah membantu, mengarahkan dan memberikan ilmu yang bermanfaat sehingga dapat menyelesaikan perkuliahan ini dengan baik.
10. Teman-teman seperjuangan dan sahabat-sahabat di kampus terima kasih atas dukungan, doa dan kebersamaan dari awal kuliah hingga selesai.
11. Semua pihak yang telah memberikan bantuan dan petunjuk sehingga saya dapat lebih mudah dalam menyusun skripsi ini.
12. Almamaterku Darmajaya yang telah memberikan pengalaman & ilmu yang berharga kepadaku.

Semoga amal perbuatan baik bapak, ibu, saudara dan semuanya mendapat balasan yang baik dari Allah SWT. Penulis menyadari skripsi ini memiliki kekurangan dan jauh dari kesempurnaan, besar harapan dari penulis untuk menerima saran dan kritik yang bersifat membangun untuk penyempurnaan di masa yang akan datang.

Akhir kata, penulis berharap skripsi ini dapat bermanfaat dan memberikan sumbangan pemikiran bagi Darmajaya dan para pembaca.

Bandar Lampung, 24 Oktober 2019  
Penulis,

**Muamar Nabila**  
**NPM. 1511050011**

## DAFTAR ISI

	<b>Halaman</b>
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>ii</b>
<b>HALAMAN PESETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>INTISARI .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>RIWAYAT HIDUP .....</b>	<b>ix</b>
<b>PRAKATA .....</b>	<b>x</b>
<b>DAFTAR ISI.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
 <b>BAB I PENDAHULUAN</b>	
1.1 .....	Latar
Belakang .....	1
1.2 .....	Peru
musan Masalah.....	3
1.3 .....	Batas
an Masalah .....	3
1.4 .....	Tujua
n Peneitian.....	3
1.5 .....	Manf
aat Penelitian .....	4
 <b>BAB II LANDASAN TEORI</b>	
2.1 Sistem .....	5
2.2 Informasi .....	5
2.3 Sistem Informasi .....	6
2.4 Basis Data.....	6
2.5 Metode Pengembangan Sistem .....	6
2.6 <i>Unified Modeling Languange (UML)</i> .....	8
2.6.1 .....	Diagr
am UML.....	10
2.7 <i>Customer Service</i> .....	17
2.8 Android.....	17
2.8.1 Android SDK .....	18
2.9 PHP ( <i>Hypertext Preprcessor</i> ).....	18
2.10 Java .....	18

2.11	MySQL .....	19
2.12	GIS ( <i>Geographic Information System</i> ) .....	19

### **BAB III METODOLOGI PENELITIAN**

3.1	Metode Pengumpulan Data.....	20
3.2	Metode Pengembangan Sistem.....	21
3.3	Alat dan Bahan Pendukung Pengembangan Sistem.....	22
3.4	Analisis Sistem.....	22
	3.4.1 Alur Sistem Berjalan pada Shoes and Care Lampung .....	22
	3.4.2 Definisi Aktor.....	23
	3.4.3 Definisi <i>Use Case</i> .....	24
	3.4.4 Skenario <i>Use Case</i> .....	24
	3.4.5 Analisis Kelemahan Sistem.....	26
3.5	<i>Use Case Diagram</i> .....	27
	3.5.1 Definisi Aktor .....	28
	3.5.2 Definisi <i>Use Case</i> .....	28
	3.5.3 Skenario <i>Use Case</i> .....	30
3.6	<i>Activity Diagram</i> .....	35
3.7	<i>Sequence Diagram</i> .....	39
	3.7.1 Penggunaan Sistem pembuatan <i>Member</i> .....	39
	3.7.2 Penggunaan Sistem Pemesanan Jasa .....	39
	3.7.3 Penggunaan Sistem Pemesanan Barang .....	40
	3.7.4 Penggunaan Sistem Admin.....	41
3.8	<i>Class Diagram</i> .....	42
3.9	Kamus Data.....	43
	3.9.1 Kamus Data <i>member_sac</i> .....	43
	3.9.2 Kamus Data Promosi .....	44
	3.9.3 Kamus Data <i>Pesan_jasa</i> .....	44
	3.9.4 Kamus Data <i>Pesan_barang</i> .....	45
	3.9.5 Kamus Data <i>Lokasi_Toko</i> .....	45
	3.9.6 Kamus Data <i>Data_Toko</i> .....	46
	3.9.7 Kamus Data <i>Produk_shoeshine</i> .....	46
	3.9.8 Kamus Data <i>member_ss</i> .....	47
	3.9.9 Kamus Data <i>member_drop</i> .....	47
3.10	Rancangan <i>Output</i> dan <i>Input</i> .....	48
	3.10.1 Rancangan <i>Output</i> .....	48
	3.10.2 Rancangan <i>Input</i> .....	50

### **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

4.1	Spesifikasi Perangkat Lunak dan Keras .....	57
4.2	Implementasi .....	58

4.2.1	Hasil Pembuatan Aplikasi.....	58
4.2.2	Tampilan Awal Aplikasi.....	58
4.2.3	Tampilan Beranda Aplikasi.....	58
4.2.4	Tampilan Toko Shoesh And Care.....	59
4.2.5	Tampilan Toko Dropit Clean.....	60
4.2.6	Tampilan Toko Shoeshine.....	60
4.2.7	Tampilan <i>Login</i> .....	61
4.2.8	Tampilan <i>Register</i> .....	62
4.2.9	Tampilan Profil Akun.....	63
4.2.10	Tampilan Pesan Jasa.....	63
4.2.11	Tampilan Menu Daftar Toko.....	64
4.2.12	Tampilan Lokasi Toko.....	65
4.2.13	Tampilan Pesan Barang.....	65
4.2.14	Tampilan Checkout.....	66
4.2.15	Tampilan Menu Pusat Bantuan.....	67
4.2.16	Tampilan Menu Promosi.....	67
4.2.17	Tampilan Halaman Admin.....	68
4.2.18	Tampilan Data Pemesanan Jasa.....	69
4.2.19	Tampilan Data Pemesanan Barang.....	69
4.3	Kelayakan Sistem.....	70

## **BAB V SIMPULAN DAN SARAN**

5.1	Simpulan.....	71
5.2	Saran.....	
	71	

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
2.2 Simbol <i>Use Case Diagram</i> .....	11
2.3 Simbol <i>Activity Diagram</i> .....	12
2.4 Simbol <i>Squence Diagram</i> .....	13
2.5 Simbol <i>Squence Diagram</i> (Lanjutan).....	14
2.6 Simbol <i>Squence Diagram</i> (Lanjutan).....	15
2.7 Simbol <i>Class Diagram</i> .....	16
2.8 Simbol <i>Class Diagram</i> (Lanjutan).....	17
3.4 Penjelasan <i>Use Case</i> Definisi Aktor .....	23
3.5 Penjelasan <i>Use Case</i> Definisi <i>Use Case</i> .....	24
3.6 Penjelasan <i>Use Case</i> Datang Ketoko .....	25
3.7 Penjelasan <i>Use Case</i> Menawarkan Jasa.....	25
3.8 Penjelasan <i>Use case</i> Menggunakan Jasa.....	25
3.9 Penjelasan <i>Use Case</i> Menerima Jasa .....	26
3.10 Penjelasan <i>Use Case</i> Memberi Nota .....	26
3.11 Analisis Kelemahan Sistem yang berjalan .....	26
3.12 Analisis Kelemahan Sistem yang berjalan (Lanjutan).....	27
3.13 Penjelasan <i>Use Case</i> Definisi Aktor .....	28
3.14 Penjelasan <i>Use Case</i> Definisi <i>Use Case</i> .....	28
3.15 Penjelasan <i>Use Case</i> Definisi <i>Use Case</i> (Lanjutan) .....	29
3.16 Skenario <i>Use Case</i> Beranda.....	30
3.17 Skenario <i>Use Case</i> Login Member .....	30
3.18 Skenario <i>Use Case</i> Logout.....	31
3.19 Skenario <i>Use Case</i> Pesan Jasa .....	31
3.20 Skenario <i>Use Case</i> Daftar Toko.....	31
3.21 Skenario <i>Use Case</i> Lihat Promosi .....	32
3.22 Skenario <i>Use Case</i> Tentang Aplikasi.....	32
3.23 Skenario <i>Use Case</i> Informasi Layanan .....	32
3.24 Skenario <i>Use Case</i> Pesan Barang .....	33
3.25 Skenario <i>Use Case</i> Lihat Map .....	33
3.26 Skenario <i>Use Case</i> Pusat Bantuan .....	33
3.27 Skenario <i>Use Case</i> Daftar Member.....	33
3.28 Skenario <i>Use Case</i> Data Pemesanan Barang .....	34
3.29 Skenario <i>Use Case</i> Data Pemesanan Jasa .....	34
3.30 Skenario <i>Use Case</i> Data Member .....	34
3.31 Skenario <i>Use Case</i> Cetak .....	35
3.32 Kamus Data Member_sac .....	43
3.33 Kamus Data Promosi.....	44
3.34 Kamus Data Pesan_Jasa.....	44
3.35 Kamus Data Pesan_Barang.....	45
3.36 Kamus Data Lokasi_Toko.....	45
3.37 Kamus Data Lokasi_Toko (Lanjutan).....	46
3.38 Kamus Data Data_Toko.....	46
3.39 Kamus Data Produk_Shoeshine .....	47

3.40 Kamus Data Member_ss .....	47
3.41 Kamus Data Member_drop .....	48
4.1 Kelayakan Sistem.....	70



## DAFTAR GAMBAR

<b>Gambar</b>	<b>Halaman</b>
2.1 Model Waterfall .....	7
3.4 <i>Use Case</i> Sistem Berjalan .....	23
3.5 <i>Use Case Diagram</i> Sistem yang diusulkan .....	27
3.6 <i>Activity Diagram</i> Pemesanan Jasa .....	35
3.7 <i>Activity Diagram</i> Pemesanan Barang .....	36
3.8 <i>Activity Diagram</i> Daftar Member .....	37
3.9 <i>Activity Diagram</i> Admin .....	38
3.10 <i>Sequence Diagram</i> Pembuatan Member .....	39
3.11 <i>Sequence Diagram</i> Pemesanan Jasa .....	40
3.12 <i>Sequence Diagram</i> Pemesanan Barang .....	41
3.13 <i>Sequence Diagram</i> Sistem Admin .....	42
3.14 <i>Class Diagram</i> Sistem Diusulkan .....	43
3.15 Rancangan <i>Output</i> Informasi Beranda .....	49
3.16 Rancangan <i>Output</i> Pesan Jasa .....	49
3.17 Rancangan Tampilan Shoes and Care .....	50
3.18 Rancangan Tampilan Daftar User .....	50
3.19 Rancangan Tampilan Login .....	51
3.20 Rancangan Tampilan Toko Shoeshine .....	51
3.21 Rancangan Tampilan Toko Dropit Clean .....	52
3.22 Rancangan Tampilan Daftar Toko .....	52
3.23 Rancangan Tampilan Pesan Barang .....	53
3.24 Rancangan Tampilan Pusat Bantuan .....	53
3.25 Rancangan Tampilan Promosi .....	54
3.26 Rancangan Tampilan Tentang Aplikasi .....	54
3.27 Rancangan Tampilan Informasi Jenis Layanan .....	55
3.28 Rancangan Tampilan Halaman Admin .....	55
3.29 Rancangan Tampilan Data Pemesanan Barang .....	56
3.30 Rancangan Tampilan Data Pemesanan Jasa .....	56
4.1 Tampilan Awal Aplikasi .....	58
4.2 Tampilan Beranda Aplikasi .....	59
4.3 Tampilan Toko Shoes And Care .....	59
4.4 Tampilan Toko Dropit Clean .....	60
4.5 Tampilan Toko Shoeshine .....	61
4.6 Tampilan Login .....	62
4.7 Tampilan Register .....	62
4.8 Tampilan Profil Akun .....	63
4.9 Tampilan Pesan Jasa .....	64
4.10 Tampilan Daftar Toko .....	64
4.11 Tampilan Lokasi Toko .....	65
4.12 Tampilan Pesan Barang .....	66
4.13 Tampilan Checkout .....	66
4.14 Tampilan Menu Pusat Bantuan .....	67
4.15 Tampilan Menu Promosi .....	68

4.16 Tampilan Halaman Admin .....	68
4.17 Tampilan Data Pemesanan Jasa .....	69
4.18 Tampilan Data Pemesanan Barang .....	69

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Sistem informasi dan teknologi komputer berkembang sangat pesat. Sejalan dengan besarnya kebutuhan terhadap informasi. Perkembangan teknologi informasi tidak lepas dari pesatnya perkembangan teknologi komputer. Karena komputer merupakan media yang dapat memberikan kemudahan bagi manusia dalam menyelesaikan pekerjaannya. Perubahan dan dinamika masyarakat yang semakin cepat seiring dengan perkembangan zaman dan teknologi sehingga memerlukan kualitas informasi yang akurat, cepat dan tepat. Dan untuk menyediakan informasi tersebut, diperlukan suatu alat bantu atau media untuk mengolah beraneka ragam data agar dapat disajikan menjadi sebuah informasi yang bermanfaat bagi penggunanya. Begitu pula pada Shoes and Care dan Shoeshine yang sangat membutuhkan bantuan teknologi komputer.

Shoes and Care Lampung didirikan pada tahun awal 2018 dan sudah berjalan selama 6 bulan dengan 4 karyawan dengan sistem kerja 2 *full time* 2 *part time* yang berlokasi di Jl. Z.A. Pagar Alam No 62, Labuhan Ratu, Kedaton, Bandar Lampung. Shoes and Care Lampung merupakan salah satu usaha penyedia layanan jasa laundry sepatu yang memberikan layanan berupa *deep cleaning, reglue & repair, repaint, unyellow bag, cap, dan fast cleaning*, kemudian adapun toko yang bergerak dibidang yang sama yaitu Shoeshine dan Dropit Clean . Shoeshine memulai karir pada awal bulan januari tahun 2019 dan memiliki 5 karyawan dengan jam kerja 09.00 pagi sampai 20.00 malam yang berlokasi di Jl. Kyai Maja No.39 A, Kedaton, Bandar Lampung. Transaksi yang dapat dihasilkan oleh Shoeshine kurang lebih 15 data per minggunya. Shoeshine memberikan layanan berupa *Fast Shine, Cleaning Shine, Leather Shine, Her Shine, Bag, Belt, Cap, Wallet, Helmet, Sandal*, serta menjual *Shoes Cleaner*. Toko Dropit Clean sudah mulai berkarir selama kurang lebih 9

bulan pada bulan Oktober 2018 yang berlokasi di Jl.H.Agus Salim no.77, Kaliawi, Bandar Lampung. Toko Dropit Clean mulai aktif bekerja di jam 10.00 pagi sampai 22.00 malam yang hanya memiliki 1 karyawan. Toko Dropit Clean menyediakan layanan cuci seperti *fast clean, deep clean, Repaint, Reglue, Unyellow, Cleaning white shoes*, dan *repair*.

Cara yang dilakukan untuk penyebaran informasi pada toko Shoes and Care Lampung, Shoeshine maupun Dropit Clean masih dilakukan melalui *social media* seperti WA (*Whatsapp*) dan Instagram, sehingga menyulitkan pengelola toko untuk menyebarkan informasi seperti layanan – layanan yang telah disediakan oleh toko dan lokasi yang masih sulit ditemukan. Proses bisnis yang terjadi di usaha Shoes and Care Lampung yaitu konsumen datang ke toko untuk menyerahkan sepatu yang akan dicuci dan menanyakan jenis layanan apa yang disediakan, lalu karyawan toko memberikan pendapat layanan jenis apa yang pantas diberikan ke konsumen untuk menggunakan jasa yang ditawarkan, kemudian pihak toko memberikan tanda bukti berupa nota kepada konsumen sebagai syarat pengambilan barang. Proses bisnis yang berjalan saat ini masih memiliki kekurangan dikarenakan pengelola toko kesulitan dalam menyebarkan informasi serta menampilkan lokasi tempat usaha dan kurangnya efisien waktu pada pelanggan dalam penggunaan layanan jasa. Hal ini menyebabkan munculnya permasalahan berupa terbatasnya penyebaran informasi, lokasi dan layanan jasa pada Shoes and Care Lampung, Shoehine dan Dropit Clean. Berdasarkan hal tersebut peneliti mempunyai ide untuk membuat aplikasi sistem informasi Shoes and Care Lampung, penyebaran informasi dapat digunakan menggunakan android dan dapat diakses oleh masyarakat luas tanpa harus mencari dan mendatangi dahulu lokasi Shoes and Care Lampung, Shoeshine dan Dropit Clean untuk menggunakan layanan jasa secara antar muka. Berdasarkan masalah tersebut penulis tertarik untuk melakukan penelitian dengan judul

**“SISTEM INFORMASI JASA CUSTOMER SERVICE PADA SHOES AND CARE LAMPUNG BERBASIS ANDROID”**, yang bertujuan membangun sebuah sistem informasi pemesanan jasa lewat android serta bisa menjadi media penyebaran informasi menu layanan, lokasi yang mudah didapatkan dan meningkatkan kinerja mutu layanan jasa yang disediakan oleh Shoes and Care Lampung, Shoeshine dan Dropit Clean.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang masalah diatas, maka dapat dirumuskan permasalahan yaitu :

1. Bagaimana cara mengatasi kesulitan dalam mencari informasi menu layanan yang disediakan Shoes and Care lampung dan lokasi yang detail?
2. Bagaimana cara melakukan pemesanan jasa dengan menggunakan aplikasi Shoes and Care Lampung berbasis android?

## **1.3 Batasan Masalah**

Batasan Masalah dibatasi pada hal berikut :

1. Sistem pemesanan layanan jasa yang akan di bangun dapat di akses menggunakan android.
2. Sistem yang dibangun membahas proses penyebaran informasi layanan, lokasi dan pemesanan jasa yang dilakukan oleh Shoes and Care Lampung , Shoeshine dan Dropit Clean.

## **1.4 Tujuan Penelitian**

Adapun tujuan penelitian yang dilakukan adalah :

1. Mempermudah penyebaran informasi menu layanan, dan promosi yang disediakan oleh Shoes and Care Lampung, Shoeshine dan Dropit Clean.
2. Meningkatkan kinerja mutu pelayanan jasa Shoes and Care Lampung dengan menggunakan android.

## **1.5 Manfaat Penelitian**

Penelitian ini memiliki beberapa manfaat diantaranya :

1. Sistem informasi yang diberikan dapat Meningkatkan kinerja mutu pelayanan jasa Shoes and Care Lampung, Shoeshine dan Dropit Clean menggunakan android.
2. Sistem informasi yang diberikan mempermudah penyebaran informasi, promosi yang disediakan oleh Shoes and Care Lampung, Shoeshine dan Dropit Clean..
3. Sistem informasi yang diberikan dapat memudahkan pelanggan mendapatkan informasi terkait jasa yang disediakan oleh Shoes and Care Lampung, Shoeshine dan Dropit Clean.
4. Sistem informasi yang diberikan memudahkan pelanggan menemukan lokasi Shoes and Care Lampung, Shoeshine dan Dropit Clean.

## **BAB II LANDASAN TEORI**

### **2.1 Sistem**

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem adalah *“Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan satu sasaran tertentu”* (Tata Sutabri, 2014).

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan operasi didalam sistem. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya, menurut Gordon B.Davis sebagai berikut : *“Sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”* (Tata Sutabri, 2014).

### **2.2 Informasi**

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu:

1. Informasi Strategis. Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.
2. Informasi Taktis. Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi Teknis. Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stock, retur penjualan, dan laporan kas harian.

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri, 2014).

### 2.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri, 2014).

### 2.4

#### Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Data Management System* (DBMS), (Rosa A.S, M.Shalahuddin, 2016).

Kebutuhan basis data dalam sistem informasi meliputi:

1. Memasukkan,  
menyimpan, dan mengambil data
2. Membuat laporan  
berdasarkan data yang telah disimpan

Tujuan dari dibuatnya table-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom, (Rosa A.S, M.Shalahuddin, 2016).

### 2.5

#### Metode

##### Pengembangan Sistem

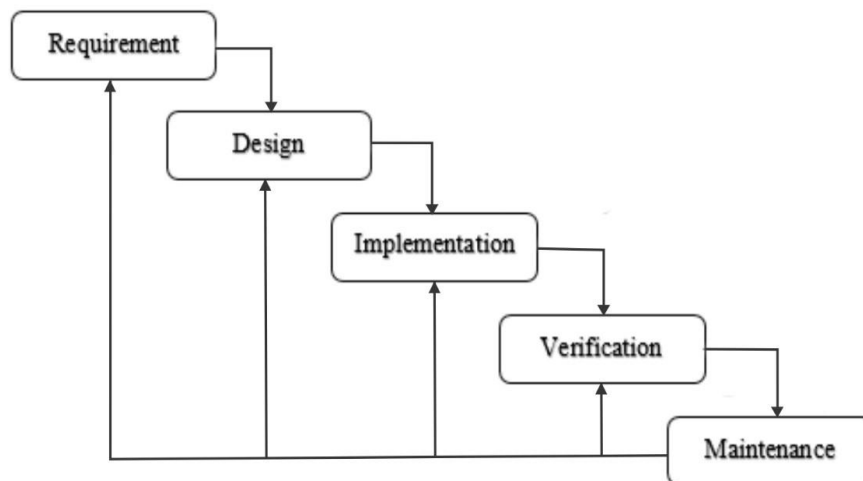
Metode pengembangan sistem sangat dibutuhkan dalam perancangan sebuah sistem karena sebelum memulai pembuatan koding – koding hendaknya merancang terlebih dahulu metode pemodelan seperti apa yang harus



digunakan dengan memprioritaskan ketepatan waktu selesai dan efektifitas dalam perancangan sebuah sistem.

(Rosa A.S, M.Shalahuddin, 2016) Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

Berikut adalah gambar model *waterfall*:



**Gambar 2.1** model *waterfall*

1. *requirement system*  
Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. *system design*  
Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak, representasi antarmuka, dan

prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. *implementation*

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. *integration & testing*

Pengujian fokus pada perangkat lunak secara logis dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. *Operation and Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

## 2.6 *Unified Modeling Language (UML)*

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang sebesar-besarnya pada developer pengembangan bahasa pemrograman berorientasi objek selanjutnya, (Rosa A.S, M.Shalahuddin, 2016).

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, CLOS. Secara actual, pengguna bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian disaat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sallay Shlaer dan Stephen Mellor (Shlaer-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon,1991), diikuti oleh Grady Booch (Booch,1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha- Premerlani-Eddy-Lorensen,1991), dan masih banyak lagi.

Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modelling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modelling Language* (UML).

Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar didalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, UML Infrastructure, UML Superstructure, dan Object Constraint Language (OCL). Seluruh spesifikasi tersebut dapat diakses diwebsite <http://www.omg.org>, (Rosa A.S, M.Shalahuddin, 2016).

### 2.6.1 Diagram UML

Untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain :

#### 1. Use Case Diagram


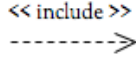
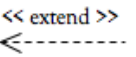
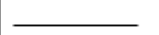


*Use case* atau diagram *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendefinisikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu, (Rosa A.S, M.Shalahuddin, 2016).

syarat penamaan pada use care adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem laen yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar kesan antar unit atau aktor.

Berikut adalah tabel simbol-simbol yang ada diagram *use case*:

**Tabel 2.2 Simbol *Use Case Diagram***

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

## 2. Activity Diagram

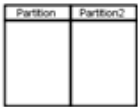





Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem, (Rosa A.S, M.Shalahuddin, 2016).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah tabel simbol-simbol yang ada pada diagram aktivitas:

**Tabel 2.3 Simbol *Activity Diagram***

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Swimlane</i>	Menunjukkan siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram.
2		<i>Action</i>	Langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meninggalkan activity, atau pada event yang spesifik.
3		<i>Initial State</i>	Menunjukkan dimana aliran kerja dimulai.
4		<i>Activity Final Node</i>	Menunjukkan dimana aliran kerja diakhiri.
5		<i>Decision Node</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dan dua atau lebih transisi sesuai dengan suatu kondisi.
6		<i>Control Flow</i>	Menunjukkan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu.



### 3. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendefinisikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstalasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

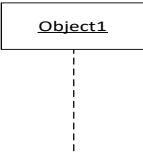

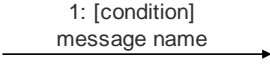
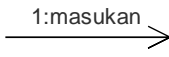
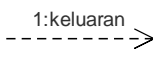
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak, (Rosa A.S, M.Shalahuddin, 2016).

Berikutnya adalah table simbol-simbol yang ada pada Sequence Diagram:

**Tabel 2.4 Simbol Sequence Diagram**

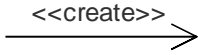
Simbol	Deskripsi
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri
Garis hidup 	Men yatakan kehidupan suatu objek.

**Tabel 2.5 Simbol Sequence Diagram (lanjutan)**

<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Arah panah mengarah pada objek yang memiliki operasi atau metode karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data /masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>

**Tabel 2.6 Simbol *Sequence Diagram* (lanjutan)**



<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
---	--

#### 4. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi, (Rosa A.S, M.Shalahuddin, 2016).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancangan dan hasil jadinya tidak sesuai.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada digram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

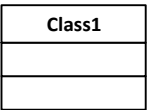
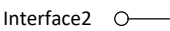
3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use care*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.


4. Kelas yang diambil dari pendefinisian data (*model*)


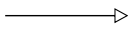
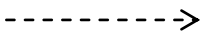

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat dibasis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggung jawabkan atau tetap ada didalam perancangan kelas.

**Tabel 2.7 Simbol Class Diagram**

Simbol	Deskripsi
	Kelas pada struktur sistem.
	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.

**Tabel 2.8 Simbol Class Diagram (lanjutan)**

	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan
---	---

	<i>multiplicity.</i>
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity.</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

## 2.5

### *Customer Service*

*Customer Service* berperan dalam memberikan pelayanan jasa yang optimal kepada pelanggan sehingga dapat memberikan kepuasan kepada pelanggan dan mempertahankan pelanggan menggunakan jasa yang ditawarkan kepada pelanggan. (Rahayu Tri Utami, 2014)

## 2.8 Android

Menurut (Nadia Firly, 2018) Android adalah sistem operasi yang berfungsi sebagai penghubung (*device*) antara pengguna dan perangkat keras pada *smarthphone* atau alat elektronik tertentu. Sehingga, hal tersebut memungkinkan pengguna dapat berinteraksi dengan *device* dan menjalankan berbagai macam aplikasi *mobile*.

### 2.8.1 Android SDK

Menurut (Nadia Firly, 2018) Android SDK adalah merupakan sebuah kit yang berfungsi untuk mengembangkan berbagai aplikasi berbasis Android oleh para *developer*. Didalam SDK telah terdapat berbagai tools yang bertujuan untuk proses pengembangan aplikasi seperti proses *debugger*, *emulator*, *software libraries*, dan dokumentasi.

## 2.9 PHP (*Hypertext Preprocessor*)

Menurut (Jubilee Enterprise, 2018 ) PHP merupakan Bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif. Dinamis artinya, website tersebut bisa berubah-ubah tampilan dan kontennya sesuai kondisi tertentu. Sebagai contoh, PHP bisa menampilkan tanggal dan hari saat ini secara berganti-ganti di dalam sebuah website. Interaktif artinya, PHP dapat memberi *feedback* bagi *user* (misalnya menampilkan hasil pencarian produk).

## 2.10 Java

*Java* menurut definisi dari *Sun Microsystem* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. *Java 2* adalah generasi kedua dari *Java Platform*. *Java* berdiri atas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine (JVM)*. JVM inilah yang akan membaca *bytecode* dalam *file .class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa *Java* disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. *Java* merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas. Saat ini *Sun Microsystem* sudah diakuisisi *Oracle Corporation* sehingga pengembangan *Java* diteruskan oleh *Oracle Corporation*(Rosa A.S, M.Shalahuddin, 2016).

### 2.11 *MySQL*

*MySQL* adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). *MySQL* juga telah mendukung bahasa pemrograman berfitur API seperti Java sehingga memudahkan para programmer java untuk berkoneksi dengan menggunakan *MySQL*. (Ochi Marshella Febriani, 2017).

### 2.12 *GIS (Geographic Information Sistem)*

SIG merupakan sebuah sistem informasi yang didesain untuk bekerja dengan sumber data spasial. SIG merupakan suatu media yang sangat handal untuk mempresentasikan data *remote Sensing* (RS) menjadi informasi yang berguna bagi banyak pihak untuk berbagai keperluan. Data spasial adalah data-data yang memiliki sistem koordinat geografis. Dengan kata lain SIG merupakan suatu sistem database yang memiliki kemampuan spesifik untuk melakukan operasi tertentu pada data. Teknologi SIG biasanya telah terintegrasi dengan teknologi database seperti *query* dan analisa statistic dengan tampilan yang unik, serta analisis geografis yang menguntungkan dengan peta. Kemampuan ini yang membuat SIG berbeda dengan sistem informasi lainnya, sehingga SIG menjadi barang berharga bagi masyarakat umum dan perusahaan untuk menjelaskan peristiwa, memprediksi pendapatan dan perencanaan strategis. SIG meliputi pemahaman tentang: pola, ruang, dan proses (metodologi) yang dibutuhkan untuk penyelesaian masalah. Selain obyektif, SIG juga dapat dijadikan sebagai alat tindakan yang sangat cepat dan efisien. SIG akan sangat berguna bila pemakai memahami konsep spasial secara utuh (Indarto, 2013).

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

Metode pengumpulan data merupakan bagian yang sangat penting dalam setiap kegiatan penelitian. Hal tersebut dilakukan untuk mendapatkan data yang akurat, terperinci, dan dapat dipercaya serta dapat dipertanggung jawabkan. Maka metode pengumpulan data harus tepat agar sesuai data yang diperlukan, didalam penelitian ini maka diperlukan beberapa tehnik pengumpulan data diantaranya yaitu:

##### **1. Observasi**

Observasi dilakukan selama 28 hari dari tanggal 28 September sampai dengan 26 Oktober 2018. Proses penelitian dilakukan dengan melakukan pengamatan langsung terhadap objek dan aktivitas dari pencucian sepatu serta melayani Konsumen pada Shoes and Care Lampung, Shoeshine dan Dropit Clean. Observasi dilakukan untuk memperoleh data dan informasi mengenai sistem yang akan dikembangkan secara detail dan akurat. Selain itu, melalui observasi juga dapat memperoleh gambaran langsung terhadap alur kerja sistem atau aktivitas sistem yang sedang berjalan secara jelas.

##### **2. Wawancara**

Wawancara dilakukan dengan teknik pengumpulan data secara tatap muka langsung dengan pihak yang bersangkutan yaitu terhadap Karyawan Shoes and Care Lampung yaitu Yayang Herdianto, Arif Muhammad selaku pemilik Shoeshine dan Rolly ruly selaku pemilik Dropit Clean.

##### **3. Studi Pustaka**

Studi pustaka diterapkan dengan melakukan penelaahan terhadap Informasi jenis layanan berupa Brosur menu yang disediakan pada Shoes and Care Lampung , Shoeshine dan Drop it Clean.

### 3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah model *waterfall* adapun tahapan yang digunakan adalah:

1. *requirement system*  
Sesuai dengan tujuan yang diharapkan, maka kontribusi yang dapat di berikan dari hasil penelitian adalah proses penyebaran informasi, lokasi dan pemesanan jasa. Penyebaran informasi yang selama ini dilakukan yaitu dengan menggunakan media sosial serta kurangnya lokasi tempat usaha. Dengan demikian untuk melakukan penyebaran informasi layanan jasa dan pemesanan jasa dilakukan dengan menggunakan *smartphone* untuk mempermudah pelanggan menggunakan jasa *customer service* berbasis android.
2. *system design*  
Pada tahapan *system design* penulis membuat rancangan *interface* dengan cara mendesain rancangan *input* dan *output* di pakai pada sistem yang diusulkan. Hal ini termasuk penggunaan metode perancangan sistem yaitu UML (*Unified Modeling Language*), struktur data, arsitektur perangkat lunak, representasi amtar muka, dan prosedur pengodean.
3. *implementation*  
Desain sistem aplikasi pemesanan jasa dilakukan dengan menggunakan aplikasi android studio dengan database MySQL.
4. *integration & testing*  
Pengujian fokus pada perangkat lunak secara dari *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 3.3 Alat Dan Bahan Pendukung Pengembangan Sistem



Untuk mendukung rancangan aplikasi yang diusulkan perlu memperhatikan beberapa hal berikut:

Perangkat keras (*hardware*) minimum yang dibutuhkan dalam pembuatan sistem informasi jasa customer service pada Shoes and Care Lampung berbasis android adalah sebagai berikut:

- 1) *Processor intel inside core15*
- 2) *RAM 4gb*
- 3) *Hardisk 1 terrabyte*
- 4) *VGA 2gb*

Perangkat lunak (*software*), minimum yang dibutuhkan dalam pembuatan sistem informasi jasa customer service pada Shoes and Care Lampung berbasis android.

a) adalah sebagai berikut:

- 1) *Operating System Windows 10*
- 2) *Android Studio*
- 3) *Database MySQL*
- 4) *Star UML*

### **3.4**

### **Analisis Sistem**

Pada tahap ini dilakukan analisis sistem berjalan pada sistem informasi jasa *customer service* pada Shoes and Care Lampung berbasis *android*.

#### **3.4.1 Alur Sistem Berjalan pada Shoes and Care Lampung**

1. Konsumen mendatangi toko untuk menyerahkan sepatu yang akan dicuci .
2. Lalu karyawan toko menjelaskan dan menawarkan layanan jenis apa yang pantas digunakan.
3. Kemudian karyawan memberikan tanda bukti berupa nota kepada konsumen sebagai syarat pengambilan barang.



	akses/lokasi yang tersedia guna mengetahui bagaimana keadaan serta layanan yang telah tersedia.
menawarkan Jasa	merupakan proses dimana karyawan toko menawarkan jasa apa yang pantas digunakan.
menggunakan Jasa	merupakan proses konsumen menggunakan jasa yang telah ditawarkan karyawan toko.
menerima Jasa	merupakan proses interaksi antara karyawan dan konsumen untuk mengetahui harga dan kesepakatan bersama.
memberi Nota	merupakan proses karyawan toko memberikan nota sebagai tanda bukti bahwa konsumen setuju mencuci sepatu.

#### 3.4.4 Skenario *Use Case*

Berikut adalah skenario jalannya masing-masing *use case* yang telah didefinisikan sebelumnya:

Nama *Use Case* : Datang ketoko.  
 Actor : Karyawan toko, Konsumen.  
 Type : *Primary Key*.  
 Tujuan : Untuk mencari lokasi toko.  
 Deskripsi : Pencarian lokasi dengan bertanya kepada orang-orang.

**Tabel 3.6 Penjelasan *Use Case* Datang ketoko**

Konsumen	Karyawan toko
----------	---------------

Menyerahkan sepatu yang akan

dicuci.

Melihat keadaan sepatu.

Nama *Use Case* : Menawarkan Jasa.  
*Actor* : Karyawan toko, Konsumen.  
*Type* : *Primary Key*.  
Tujuan : Menawarkan layanan apa yang pantas digunakan oleh konsumen.  
Deskripsi : Karyawan menjelaskan layanan dan harga yang pantas digunakan.

### **Tabel 3.7 Penjelasan Use Case Menawarkan Jasa**

Konsumen

Karyawan toko

Memberi layanan yang tepat.

Setuju dengan layanan yang tepat.

Nama *Use Case* : Menggunakan Jasa.  
*Actor* : Karyawan toko, Konsumen.  
*Type* : *Primary Key*.  
Tujuan : Untuk memberikan saran layanan yang tepat.  
Deskripsi : Karyawan menjelaskan layanan dan harga yang pantas digunakan.

### **Tabel 3.8 Penjelasan Use Case Menggunakan Jasa**

Konsumen

Karyawan toko

. Menggunakan jasa yang disarankan.

Menerima jasa.

Nama *Use Case* : Menerima Jasa.  
*Actor* : Karyawan toko, Konsumen.  
*Type* : *Primary Key*.  
Tujuan : Untuk mengetahui harga dan kesepakatan antara konsumen dan karyawan toko.  
Deskripsi : Karyawan menjelaskan layanan dan harga

yang telah disepakati.

**Tabel 3.9 Penjelasan Use Case Menerima Jasa**

Konsumen	Karyawan toko
Menerima jasa yang digunakan.	
Menunggu waktu sepatu kapan selesai.	

Nama Use Case	: Memberi Nota.
Actor	: Karyawan toko, Konsumen.
Type	: Primary Key.
Tujuan	: Untuk pengambilan barang.
Deskripsi	: Karyawan memberikan Nota kepada konsumen sebagai tanda bukti.

**Tabel 3.10 Penjelasan Use Case Memberi Nota**

Konsumen	Karyawan toko
Memberi Nota	
Menunggu waktu sepatu kapan selesai	

**3.4.5 Analisis Kelemahan Sistem**

**Tabel 3.11 Analisis Kelemahan Sistem yang berjalan**

Permasalahan	Pemecahan masalah
kurangnya penyebaran informasi layanan-layanan yang disediakan oleh Shoes and Care Lampung.	Membuat sistem informasi jasa <i>customer service</i> pada Shoes and Care Lampung berbasis android.

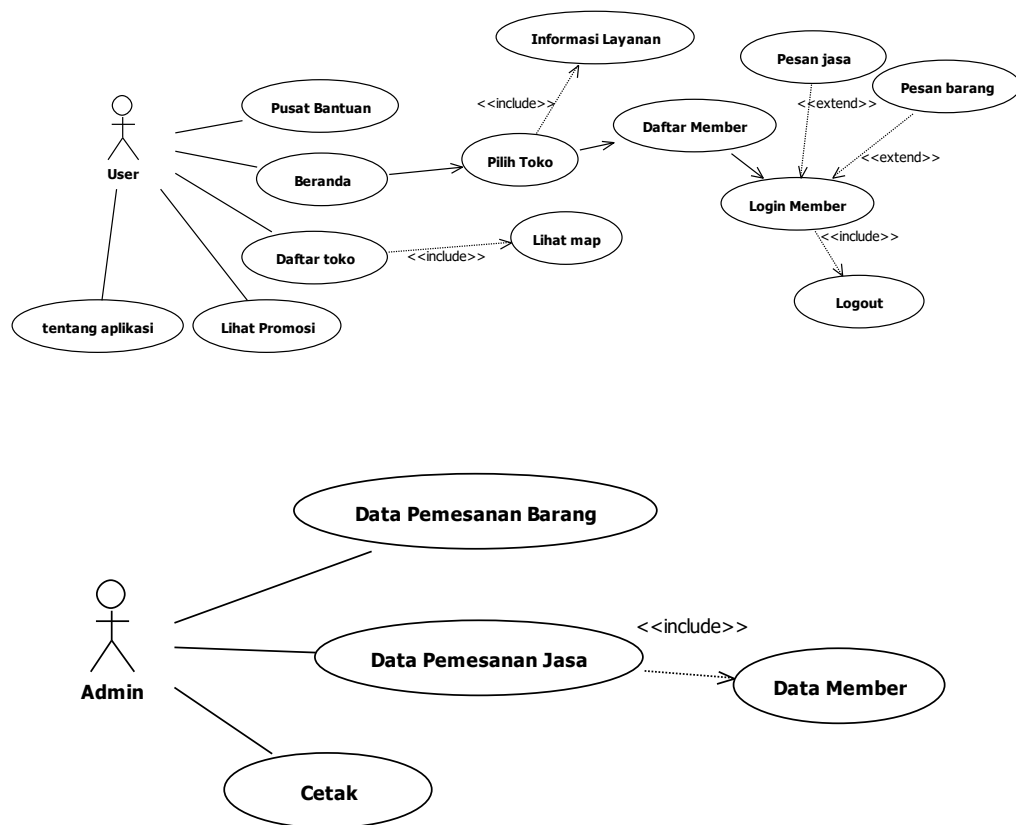
**Tabel 3.12 Analisis Kelemahan Sistem yang berjalan (lanjutan)**

konsumen tidak dapat Membuat sistem informasi mengenai mengetahui akses jalur untuk peta yang disediakan pada aplikasi menuju alamat toko. untuk mencari alamat dengan mudah.

Konsumen tidak dapat Melakukan pemesanan jasa dapat menggunakan jasa yang telah secara *online*. disediakan oleh toko.

terbatasnya penyebaran promosi Membuat sistem informasi yang akan menampilkan promosi masing-masing toko.

### 3.5 Use Case Diagram



Gambar 3.5 Use Case Diagram Sistem yang diusulkan

#### 3.5.1 Definisi aktor

Berikut adalah deskripsi pendefinisian aktor pada sistem informasi jasa *customer service* pada Shoes and Care Lampung:

**Tabel 3.13 Penjelasan Use Case Definisi aktor**

No	Aktor	Deskripsi
	ser	rang yang menggunakan jasa laundry.
	dmin	rang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelola data pemesanan.

### 3.5.2 Definisi Use Case

Berikut adalah deskripsi pendefinisian *use case* pada sistem informasi jasa *customer service* pada Shoes and Care Lampung:

**Tabel 3.14 Penjelasan Use Case Definisi Use Case**

No	Use Case	Deskripsi
	ogin Member	erupakan proses dimana konsumen memasuki aplikasi.
	aftar Member	erupakan proses dimana konsumen menjadi member pada toko.
	ogout	erupakan proses dimana konsumen keluar dari akun.
	eranda	erupakan proses dimana konsumen mengetahui halaman keterangan singkat fungsi aplikasi dan daftar toko .
	iformasi Layanan	erupakan proses dimana konsumen melihat jenis layanan serta harga yang

disediakan oleh toko.

**Tabel 3.15 Penjelasan Use Case Definisi Use Case (lanjutan)**

1	Pemesanan Jasa	Merupakan proses dimana konsumen dapat memesan jasa <i>shoes laundry</i> .
2	Pemilihan Toko	Merupakan proses dimana konsumen dapat memilih toko yang tersedia.
3	Pemilihan Toko	Merupakan proses dimana konsumen dapat melihat toko yang tersedia.
4	Pemilihan Map	Merupakan proses dimana konsumen dapat mengetahui lokasi toko.
5	Pemesanan Barang	Merupakan proses dimana konsumen dapat memesan barang di toko Shoeshine.
6	Pusat Bantuan	Merupakan proses dimana konsumen dapat melihat informasi pada aplikasi.
7	Pemilihan Promosi	Merupakan proses dimana konsumen dapat mengetahui promo yang tersedia oleh toko.
8	Peringatan aplikasi	Merupakan dimana konsumen dapat melihat informasi versi aplikasi yang dibuat.
9	Statistik Pemesanan Barang	Merupakan dimana Admin dapat mengetahui siapa saja yang memesan Barang.
10	Statistik Pemesanan Jasa	Merupakan dimana Admin dapat mengetahui siapa saja yang memesan



Jasa.

5     ata Member           lerupakan dimana Admin dapat mengetahui siapa saja member yang telah memesan jasa.

7     etak                   lerupakan dimana Admin Mencetak hasil dari data pemesanan.

### 3.5.3 Skenario Use Case

Berikut adalah skenario jalannya masing-masing *use case* yang telah didefinisikan sebelumnya:

Nama *Use Case*       : Beranda  
Actor                   : User.  
Type                    : *Primary Key*.  
Tujuan                  : Untuk melihat tampilan beranda aplikasi.  
Deskripsi               : User dapat melihat tampilan beranda aplikasi.

**Tabel 3.16 Skenario Use Case Beranda**

User
1. Membuka aplikasi
2. Melihat tampilan awal aplikasi

Nama *Use Case*       : Login Member  
Actor                   : User.  
Type                    : *Primary Key*.  
Tujuan                  : Untuk masuk ke aplikasi.  
Deskripsi               : User memasukkan *username* dan *Password*.

**Tabel 3.17 Skenario Use Case Login Member**

User
1. User menekan tombol login.
2. User memasukkan <i>username</i> dan <i>password</i> .

Nama *Use Case*       : Logout

*Actor* : User.  
*Type* : *Primary Key*.  
 Tujuan : Untuk keluar dari akun.  
 Deskripsi : User dapat keluar akun dengan menekan tombol logout.

**Tabel 3.18 Skenario Use Case Logout**

User
1. User dapat keluar dengan menekan tombol logout.

*Nama Use Case* : Pesan Jasa  
*Actor* : User.  
*Type* : *Primary Key*.  
 Tujuan : Untuk memesan jasa *shoes laundry*.  
 Deskripsi : User dapat memesan jenis layanan yang telah disediakan oleh toko.

**Tabel 3.19 Skenario Use Case Pesan Jasa**

User
1. Memilih menu Pesan Jasa.
2. Mengisi form pemesanan Jasa.

*Nama Use Case* : Daftar Toko  
*Actor* : User.  
*Type* : *Primary Key*.  
 Tujuan : Untuk melihat Daftar Toko.  
 Deskripsi : User dapat melihat daftar toko yang sudah terdaftar.

**Tabel 3.20 Skenario Use Case Daftar Toko**

User
1. User dapat melihat daftar toko yang sudah terdaftar.

*Nama Use Case* : Lihat Promosi  
*Actor* : User.  
*Type* : *Primary Key*.  
 Tujuan : Untuk melihat Promosi yang tersedia.  
 Deskripsi : User dapat melihat Promosi yang sudah

tersedia.

**Tabel 3.21 Skenario Use Case Lihat Promosi**

User
1. User dapat melihat Promosi yang sudah tersedia.

Nama *Use Case* : Tentang aplikasi  
*Actor* : User.  
*Type* : *Primary Key*.  
Tujuan : Untuk melihat tentang informasi yang membuat aplikasi.  
Deskripsi : User membuka menu informasi tentang aplikasi tersebut.

**Tabel 3.22 Skenario Use Case Tentang aplikasi**

User
1. User dapat melihat informasi aplikasi tersebut.

Nama *Use Case* : Informasi Layanan  
*Actor* : User.  
*Type* : *Primary Key*.  
Tujuan : Untuk melihat tentang informasi jenis Layanan yang disediakan oleh toko.  
Deskripsi : User membuka menu informasi jenis layanan serta harga yang disediakan oleh toko.

**Tabel 3.23 Skenario Use Case Informasi Layanan**

User
1. User dapat melihat informasi jenis layanan serta harga yang telah disediakan oleh toko.

Nama *Use Case* : Pesan Barang  
*Actor* : User.  
*Type* : *Primary Key*.  
Tujuan : Untuk melakukan transaksi pembelian barang yang disediakan oleh toko.  
Deskripsi : User memilih barang yang ingin dibeli.

**Tabel 3.24 Skenario Use Case Pesan Barang**

User
1. Untuk melakukan transaksi pembelian barang yang disediakan oleh toko.

Nama Use Case : Lihat Map  
Actor : User.  
Type : *Primary Key*.  
Tujuan : Untuk melihat lokasi toko.  
Deskripsi : User dapat menemukan lokasi toko.

**Tabel 3.25 Skenario Use Case Lihat Map**

User
1. Untuk melihat lokasi toko.

Nama Use Case : Pusat Bantuan  
Actor : User.  
Type : *Primary Key*.  
Tujuan : Untuk informasi bantuan pada aplikasi.  
Deskripsi : User dapat mengetahui penjelasan pada aplikasi.

**Tabel 3.26 Skenario Use Case Pusat Bantuan**

User
1. Untuk mengetahui penjelasan pada aplikasi

Nama Use Case : Daftar Member  
Actor : User.  
Type : *Primary Key*.  
Tujuan : Untuk pendaftaran menjadi member.  
Deskripsi : User dapat menjadi member pada toko.

**Tabel 3.27 Skenario Use Case Daftar Member**

User
1. User terdaftar menjadi member

Nama Use Case : Data Pemesanan Barang  
Actor : Admin.  
Type : *Primary Key*.  
Tujuan : Untuk mengetahui data pemesan barang.  
Deskripsi : Admin mengetahui data pemesanan barang.

**Tabel 3.28 Skenario Use Case Data Pemesanan Barang**

Admin
1. Admin mengetahui Data Pemesanan Barang

Nama Use Case : Data Pemesanan Jasa  
Actor : Admin.  
Type : *Primary Key*.  
Tujuan : Untuk mengetahui data pemesan jasa.  
Deskripsi : Admin mengetahui data pemesanan jasa.

**Tabel 3.29 Skenario Use Case Data Pemesanan Jasa**

Admin
1. Admin mengetahui Data Pemesanan Jasa

Nama Use Case : Data Member  
Actor : Admin.  
Type : *Primary Key*.  
Tujuan : Untuk mengetahui data member pada toko.  
Deskripsi : Admin mengetahui data member pada toko.

**Tabel 3.30 Skenario Use Case Data Member**

Admin
1. Admin mengetahui Data Member

Nama Use Case : Cetak  
Actor : Admin.  
Type : *Primary Key*.  
Tujuan : Untuk mencetak laporan data pemesanan.  
Deskripsi : Admin mengetahui laporan data pemesanan..

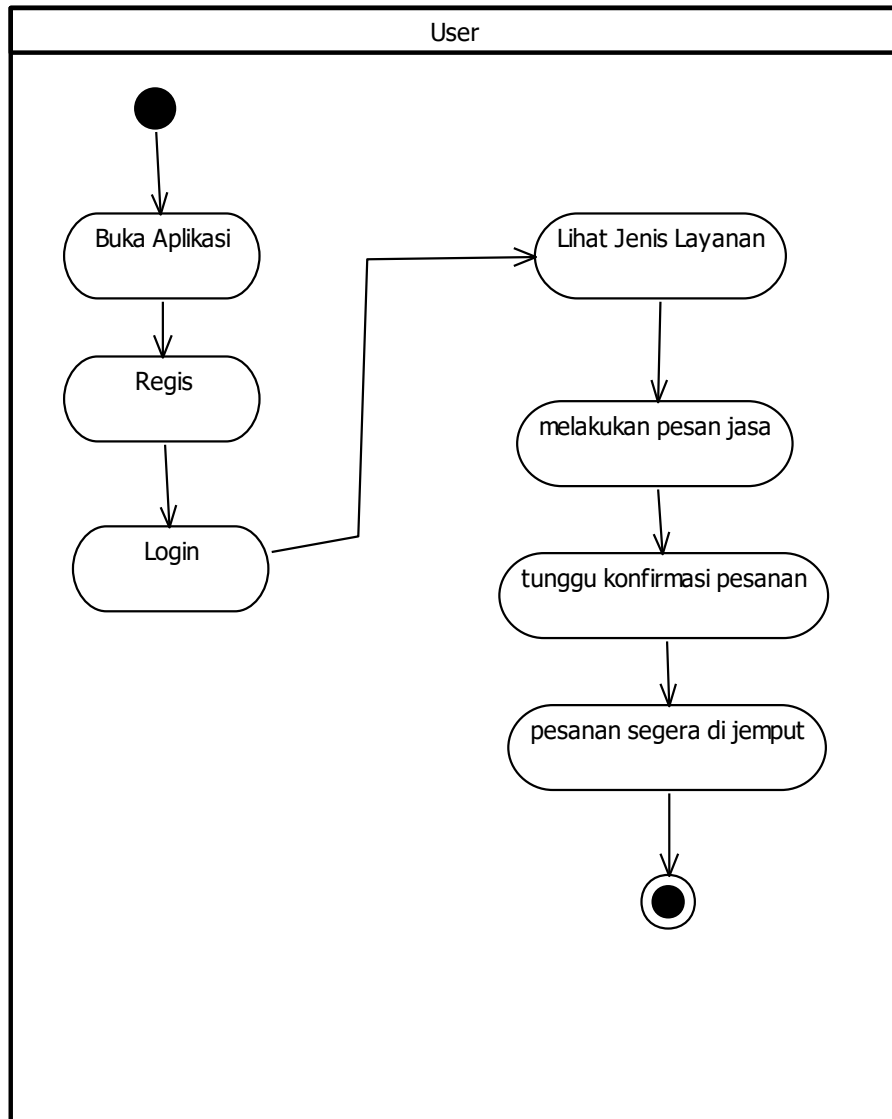
**Tabel 3.31 Skenario Use Case Cetak**

Admin
1. Admin mencetak laporan data pemesnan.

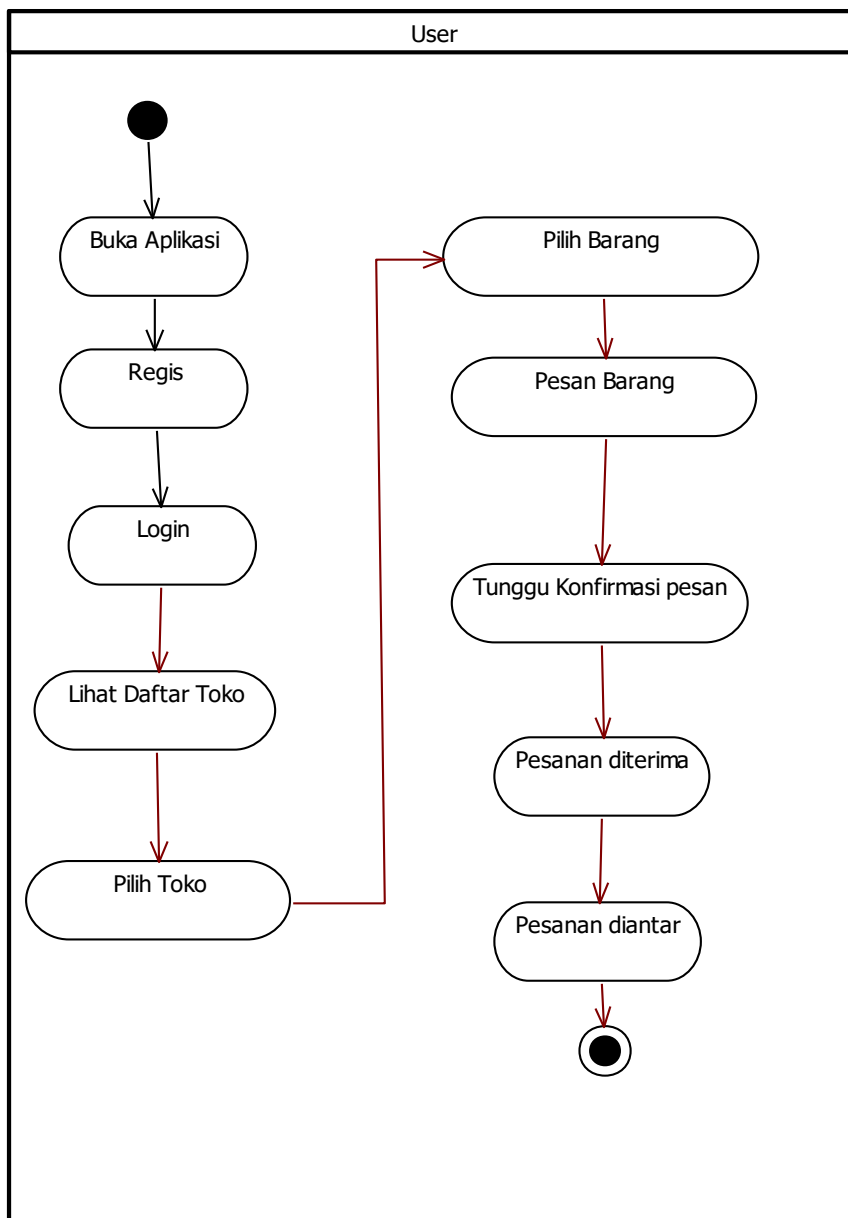
### 3.6 Activity Diagram

Adapun desain sistem yang diusulkan dapat dijelaskan menggunakan

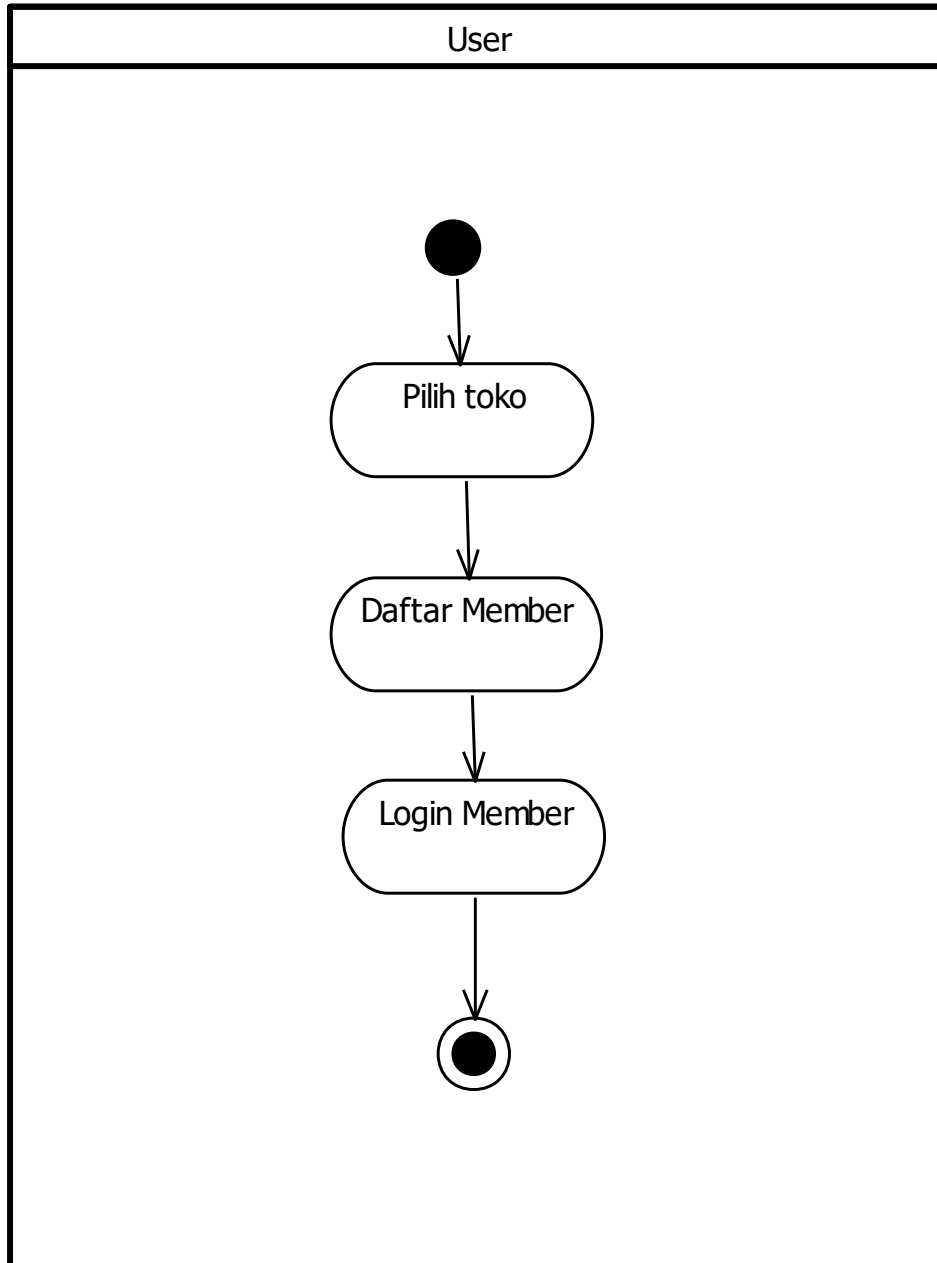
*Activity diagram* seperti pada gambar 3.6 berikut :



**Gambar 3.6 Activity Diagram Pemesanan Jasa**

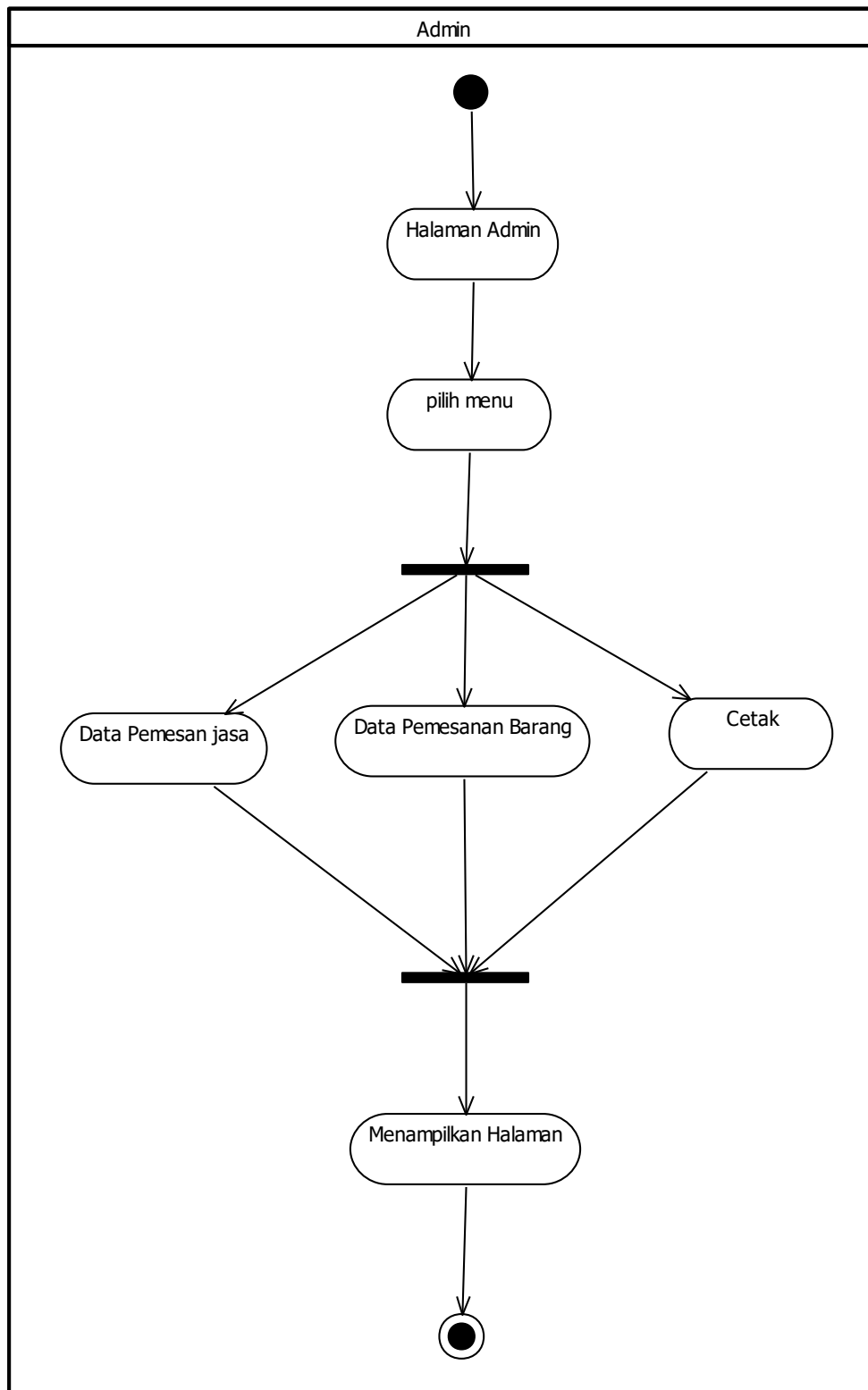


**Gambar 3.7 Activity Diagram Pemesanan Barang**



**Gambar 3.8** *Activity Diagram* Daftar Member



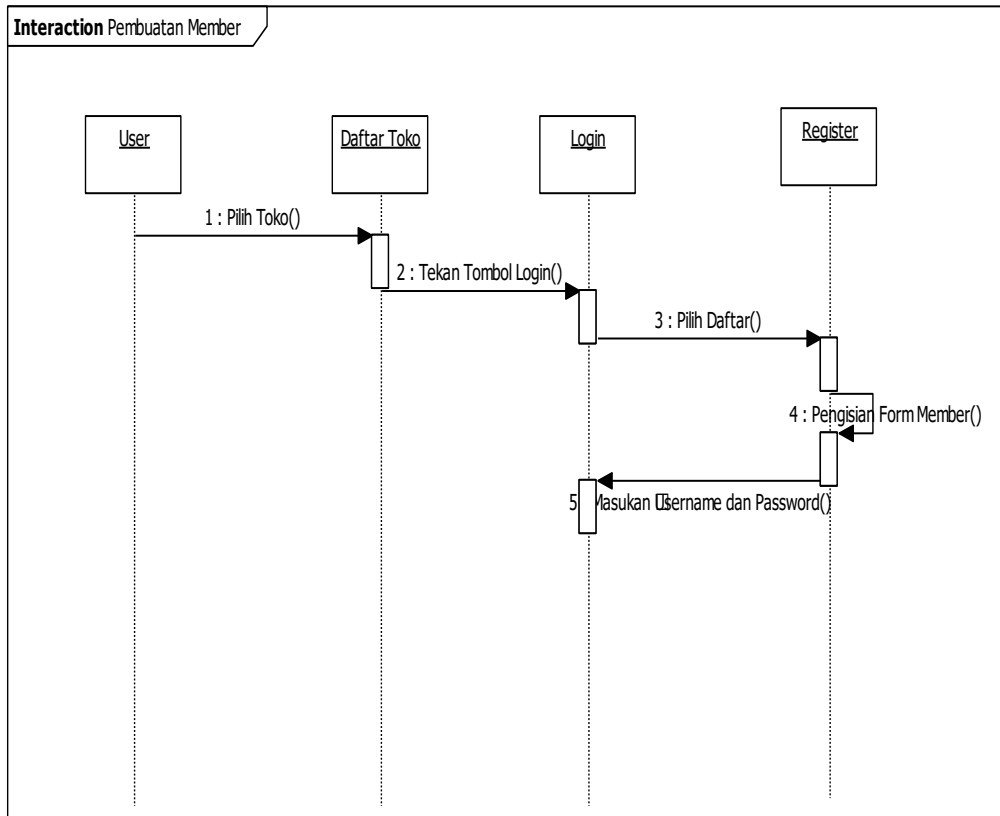


**Gambar 3.9 Activity Diagram Admin**

**3.7 Sequence Diagram**

**3.7.1 Penggunaan Sistem pembuatan Member**

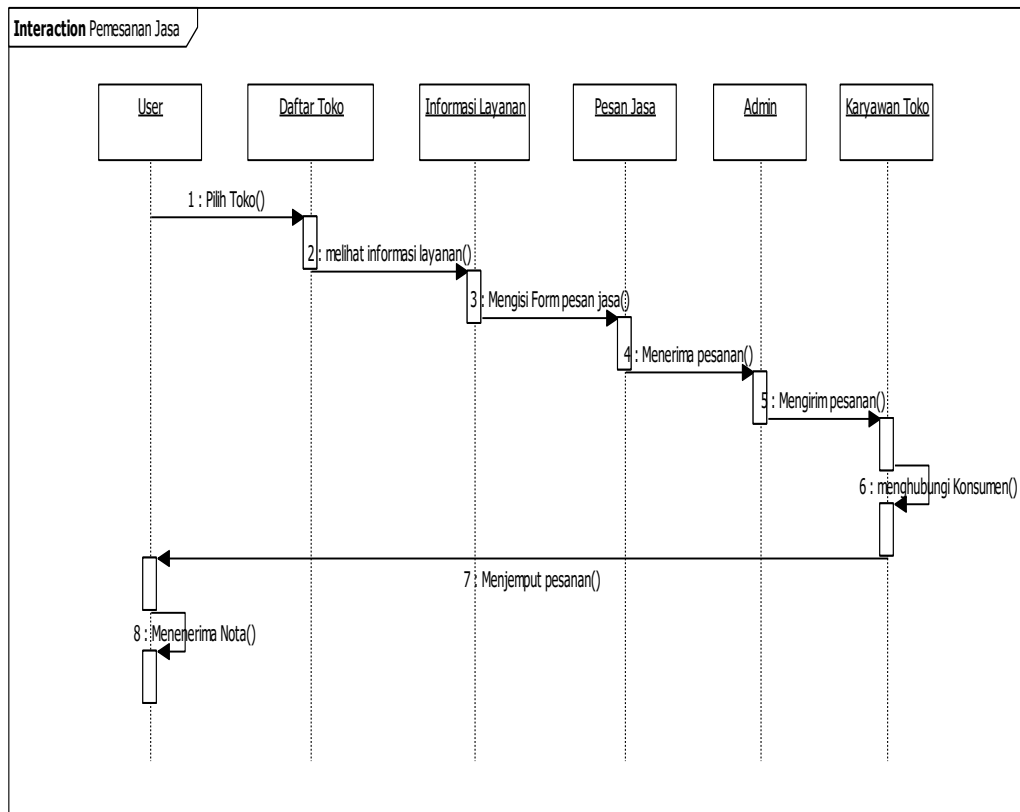
Desain penggunaan sistem yang diusulkan dapat dijelaskan menggunakan *Sequence diagram* pembuatan member seperti pada gambar 3.10 berikut :



**Gambar 3.10** *Sequence Diagram* Pembuatan Member

### 3.7.2 Penggunaan Sistem Pemesanan Jasa

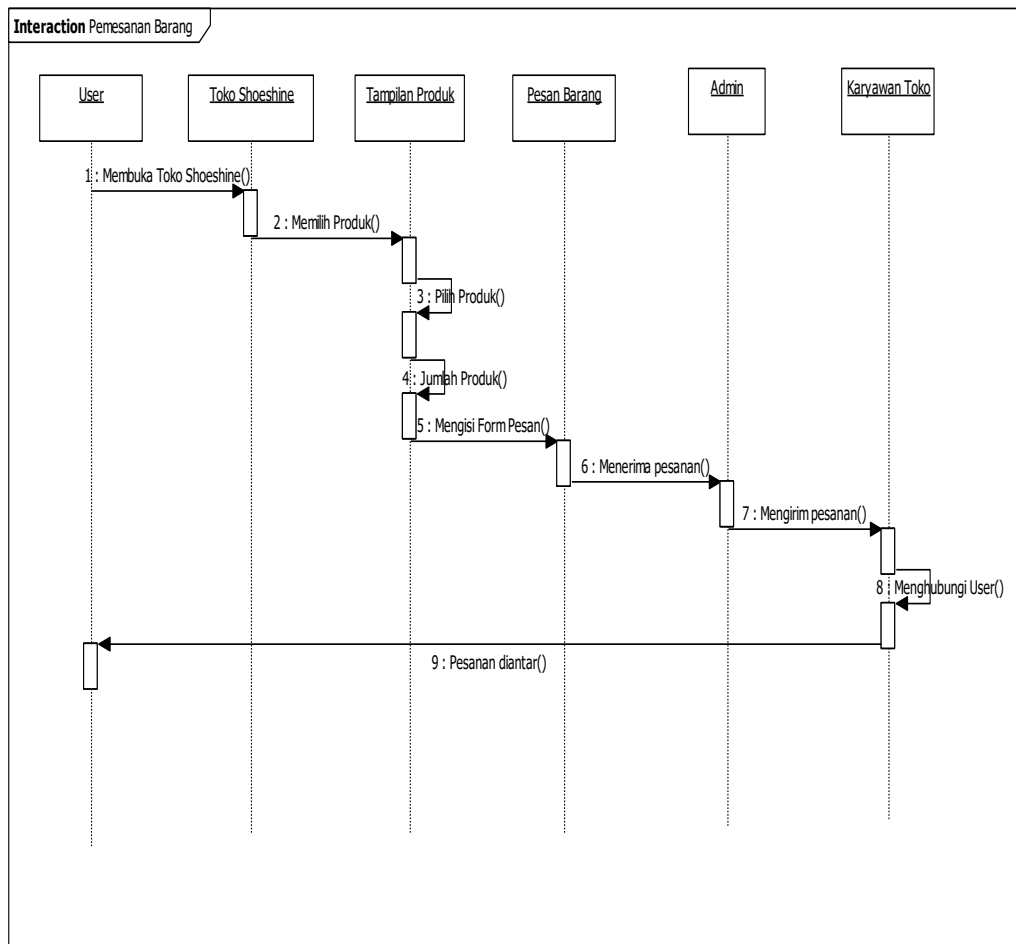
Adapun desain penggunaan sistem aplikasi yang diusulkan dapat dijelaskan menggunakan *Sequence diagram* penggunaan sistem pemesanan jasa seperti pada gambar 3.11 berikut :



**Gambar 3.11 Sequence Diagram Pemesanan Jasa**

### 3.7.3 Penggunaan Sistem Pemesanan Barang

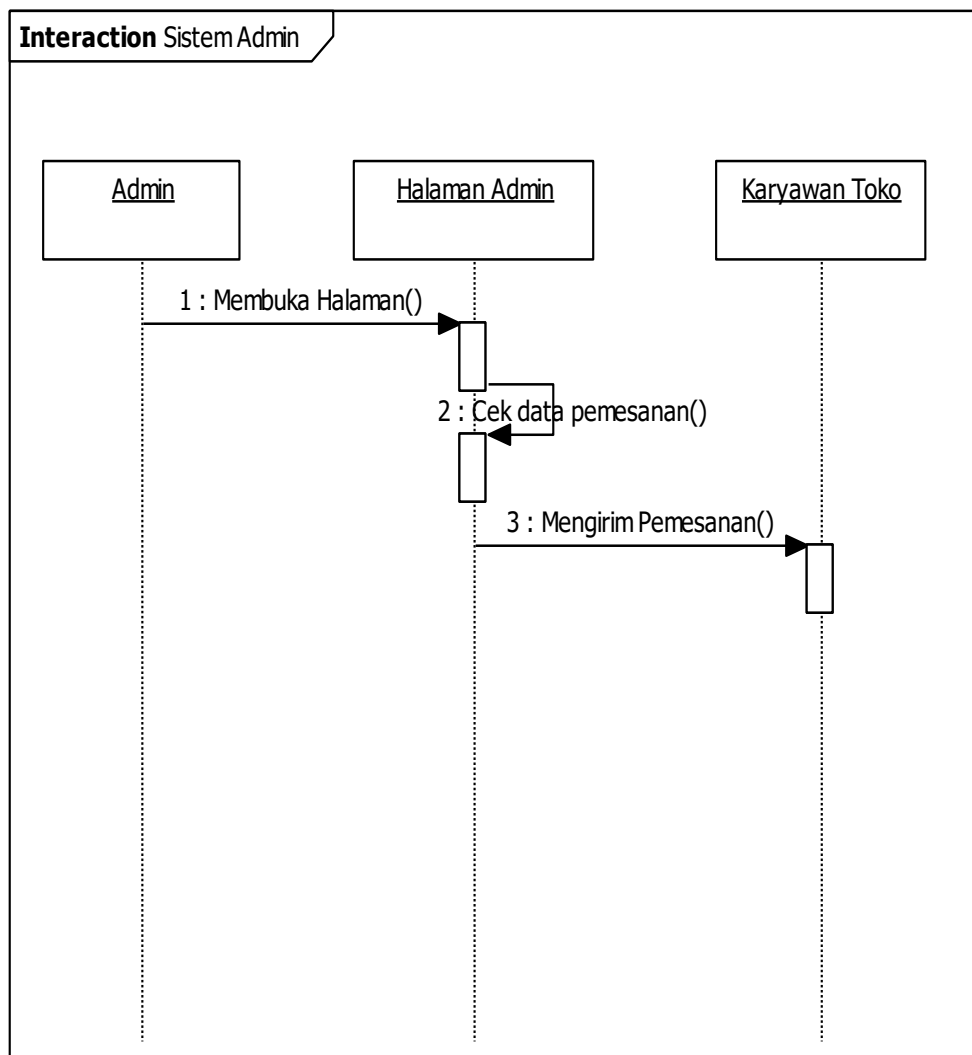
Adapun desain penggunaan sistem aplikasi yang diusulkan dapat dijelaskan menggunakan *Sequence diagram* penggunaan sistem pemesanan barang seperti pada gambar 3.12 berikut :



**Gambar 3.12 Sequence Diagram Pemesanan Barang**

### 3.7.4 Penggunaan Sistem Admin

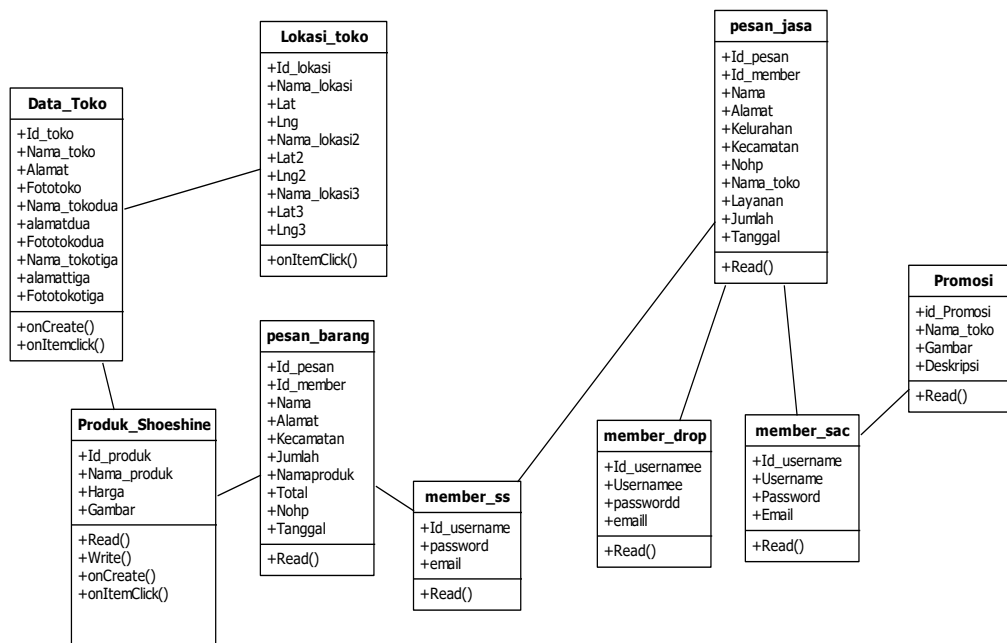
Adapun desain penggunaan sistem aplikasi yang diusulkan dapat dijelaskan menggunakan *Sequence diagram* penggunaan sistem Admin seperti pada gambar 3.13 berikut :



**Gambar 3.13 Sequence Diagram Sistem Admin**

### 3.8 Class Diagram

Adapun desain aplikasi yang diusulkan dapat dijelaskan menggunakan *class diagram*, seperti pada gambar 3.14 berikut :



**Gambar 3.14 Class Diagram Sistem Diusulkan**

### 3.9 Kamus Data

#### 3.9.1 Kamus Data member\_sac

Nama Tabel : member\_sac  
 Primary Key : id\_username  
 Size : 3

**Tabel 3.32 Kamus Data member\_sac**

Field Name	Type	Size	Description
Id_Username	Int	3	id_Konsumen
Username	Varchar	10	Nama Pengguna
Password	Varchar	10	Password
Email	Varchar	15	Email pengguna

#### 3.9.2 Kamus Data Promosi

Nama Tabel : Promosi  
 Primary Key : id\_promosi  
 Size : 3

**Tabel 3.33 Kamus Data promosi**

Field Name	Type	Size	Description
Id_promosi	Int	3	id_promosi
Nama_toko	Varchar	10	Nama Toko
Gambar	Text	-	Gambar Promosi
Deskripsi	Text	-	Penjelasan Promosi

**3.9.3 Kamus Data Pesan\_jasa**

Nama Tabel : Pesan\_jasa

*Primary Key* : id\_pesanan

Size : 3

**Tabel 3.34 Kamus Data Pesan\_jasa**

Field Name	Type	Size	Description
Id_pesanan	Int	3	id_pesanan
Id_member	Int	3	Id_member
Nama	Varchar	15	Nama Konsumen
Alamat	Text	-	Alamat Konsumen
Kelurahan	Varchar	10	Kelurahan Konsumen
Kecamatan	Varchar	20	Kecamatan Konsumen
NoHp	Varchar	15	Nohp konsumen
Nama_toko	Varchar	17	Nama toko
Layanan	Varchar	15	Layanan toko
Jumlah	Int	2	Jumlah Barang
Tanggal	Date	-	Tanggal pemesanan

**3.9.4 Kamus Data Pesan\_barang**

Nama Tabel : Pesan\_barang

*Primary Key* : id\_pesanan

Size : 3

**Tabel 3.35 Kamus Data Pesan\_barang**

Field Name	Type	Size	Description
Id_pesan	Int	3	id_pesan
Id_member	Int	3	id_member
Nama	Varchar	15	Nama Konsumen
Alamat	Text	-	Alamat Konsumen
Kecamatan	Varchar	20	Kecamatan Konsumen
Namaproduk	Varchar	15	Nama Produk
Jumlah	Varchar	2	Jumlah produk
Total	Varchar	10	Total Belanja
NoHp	Varchar	15	Nohp konsumen
Tanggal	Date	-	Tanggal pemesanan

**3.9.5 Kamus Data Lokasi\_Toko**

Nama Tabel : Lokasi\_toko

Primary Key : id\_lokasi

Size : 3

**Tabel 3.36 Kamus Data Lokasi\_Toko**

Field Name	Type	Size	Description
Id_lokasi	Int	3	id_lokasi
Nama_lokasi	Varchar	15	Nama toko
Lat	Double	-	Kordinat toko
Lng	Double	-	Kordinat toko
Nama_lokasi2	Varchar	13	Nama toko

**Tabel 3.37 Kamus Data Lokasi\_Toko (lanjutan)**

Lat2	Double	-	Kordinat toko
Lng2	Double	-	Kordinat toko
nama_toko3	archar	)	nama toko





Gambar	Text	-	Gambar produk
--------	------	---	---------------

### 3.9.8 Kamus Data member\_ss

Nama Tabel : member\_ss  
*Primary Key* : id\_username  
 Size : 3

**Tabel 3.40 Kamus Data member\_ss**

Field Name	Type	Size	Description
Id_Username	Int	3	id_Konsumen
Username	Varchar	10	Nama Pengguna
Password	Varchar	10	Password
Email	Varchar	15	Email pengguna

### 3.9.9 Kamus Data member\_drop

Nama Tabel : member\_drop  
*Primary Key* : Id\_username  
 Size : 3

**Tabel 3.41 Kamus Data member\_drop**

Field Name	Type	Size	Description
Id_Usernamee	Int	3	id_Konsumen
Usernamee	Varchar	10	Nama Pengguna
Passwordd	Varchar	10	Password
Email	Varchar	15	Email pengguna

### 3.10 Rancangan *Output* dan *Input*

Berikut rancangan *output* dan *input* dalam sistem informasi jasa *customer service* pada Shoes And Care Lampung.

### **3.10.1 Rancangan *Output***

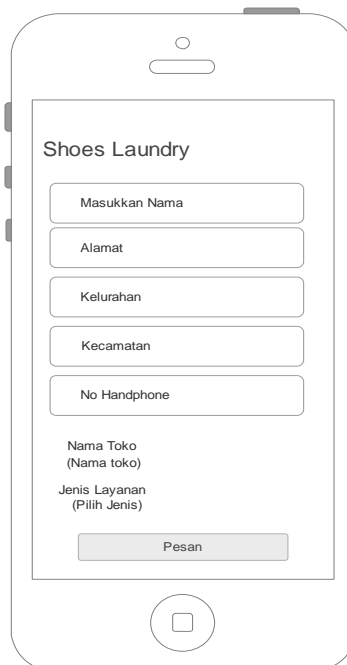
Rancangan keluaran (*Design Output*) adalah informasi yang dihasilkan oleh proses pengolahan data yang dilakukan secara otomatis oleh aplikasi yang telah terkomputerisasi. Rancangan keluaran (*Design Output*) ini merupakan bentuk dan format yang dihasilkan oleh program. Berikut rancangan keluaran (*Design Output*) yang digunakan dalam sistem informasi jasa *customer service* pada Shoes And Care Lampung. Berikut merupakan rancangan output yang diusulkan.

#### **3.10.1.1 Rancangan *Output* Informasi Beranda**



**Gambar 3.15 Rancangan *Output* Informasi Beranda**

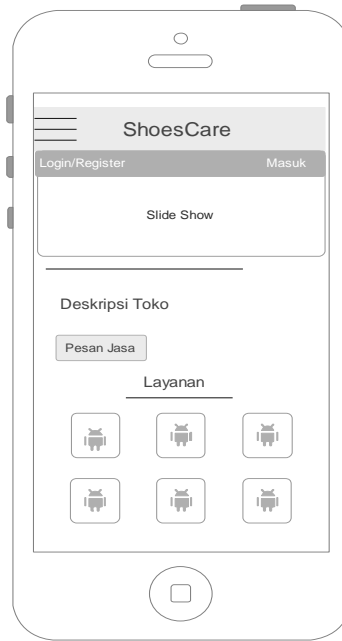
### **3.10.1.2 Tampilan *Output* Pesan Jasa**



**Gambar 3.16 Rancangan *Output* Pesan Jasa**

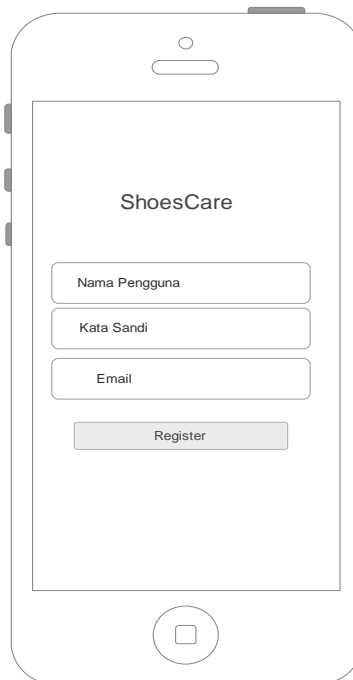
### **3.10.2 Rancangan *Input***

#### **3.10.2.1 Tampilan Aplikasi Shoes and Care**



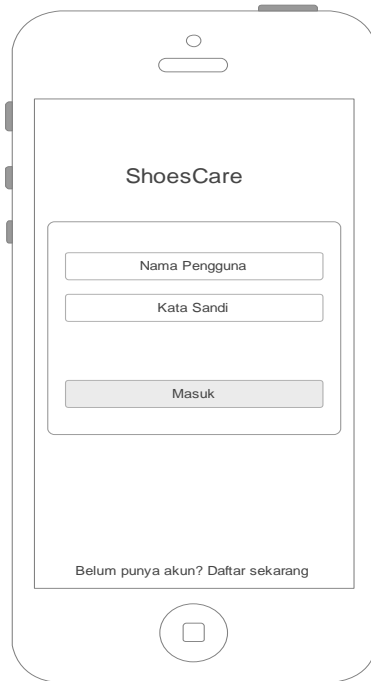
**Gambar 3.17 Rancangan Tampilan Shoes and Care**

### 3.10.2.2 Tampilan Daftar User



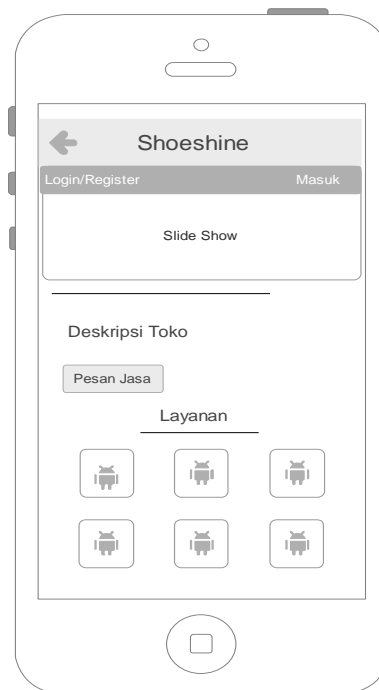
**Gambar 3.18 Rancangan Tampilan Daftar User**

### 3.10.2.3 Tampilan *Login* Aplikasi



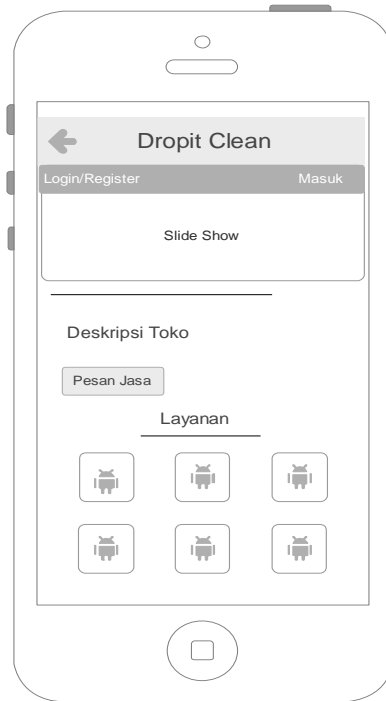
**Gambar 3.19 Rancangan Tampilan *Login***

#### **3.10.2.4 Tampilan Toko Shoeshine**



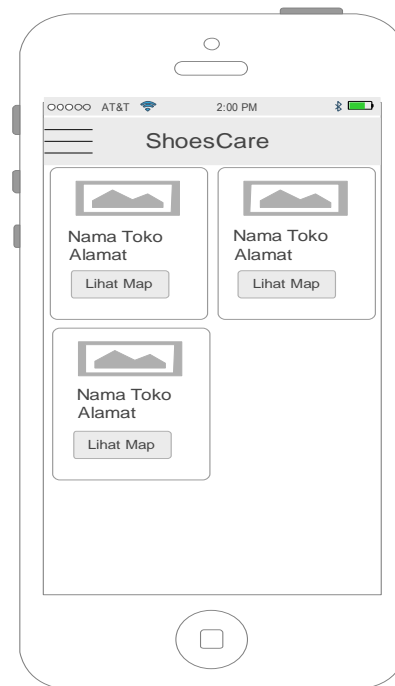
**Gambar 3.20 Rancangan Tampilan Toko Shoeshine**

#### **3.10.2.6 Tampilan Toko Dropit Clean**



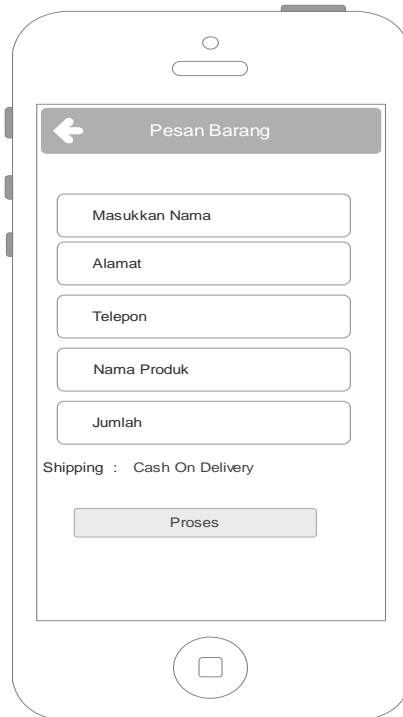
**Gambar 3.21 Rancangan Tampilan Toko Dropit Clean**

### **3.10.2.8 Tampilan Daftar Toko**



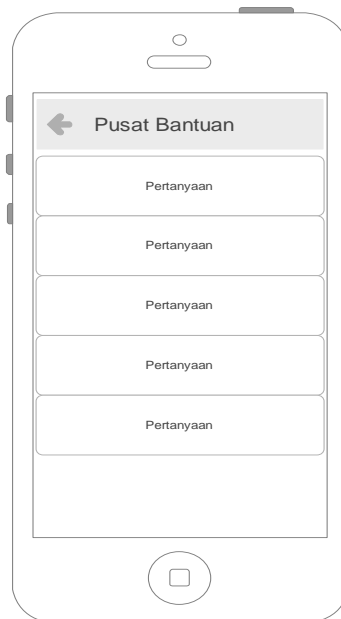
**Gambar 3.22 Rancangan Tampilan Daftar Toko**

### **3.10.2.9 Tampilan Pesan Barang**



**Gambar 3.23 Rancangan Tampilan Pesan Barang**

### **3.10.2.10 Tampilan Pusat Bantuan**

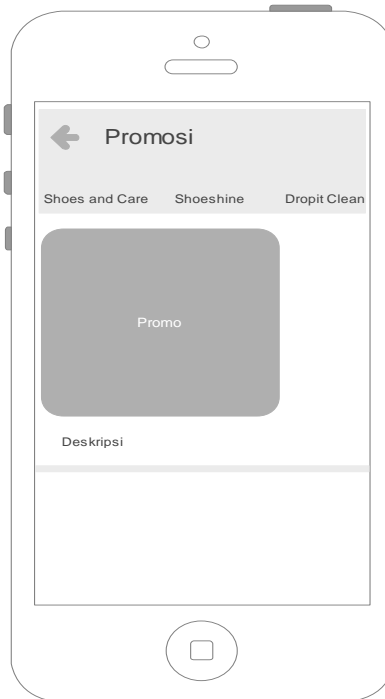


**Gambar 3.24 Rancangan Tampilan Pusat Bantuan**

### **3.10.2.11**

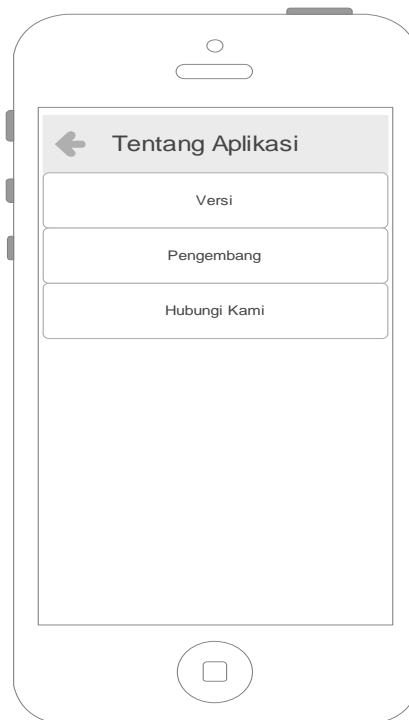
### **Tampilan Pomosi**





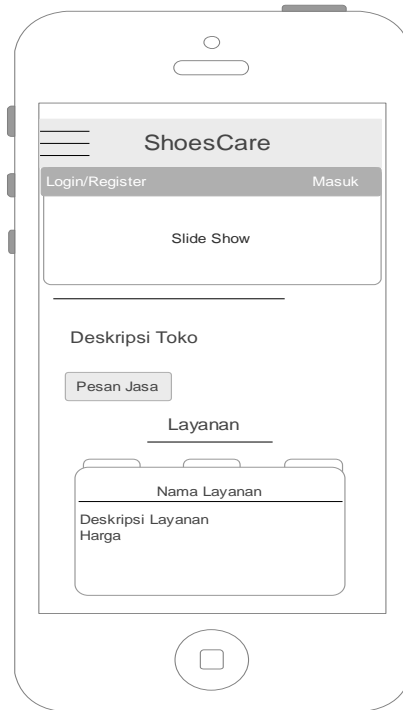
**Gambar 3.25 Rancangan Tampilan Promosi**

### **3.10.2.12 Tampilan Tentang Aplikasi**



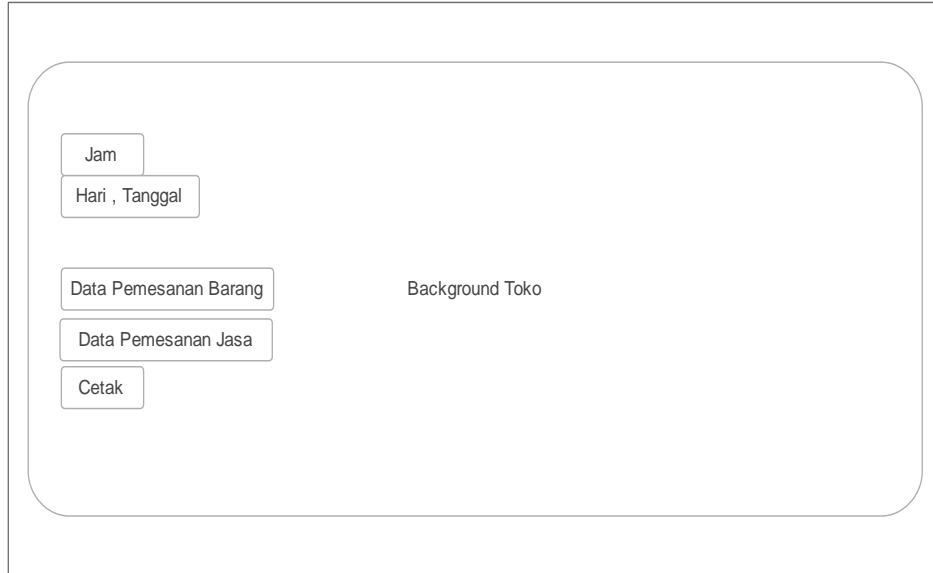
**Gambar 3.26 Rancangan Tampilan Tentang Aplikasi**

### **3.10.2.13 Tampilan Informasi Jenis Layanan**



**Gambar 3.27 Rancangan Tampilan Informasi Jenis Layanan**

**3.10.2.14 Tampilan Halaman Admin**



**Gambar 3.28 Rancangan Tampilan Halaman Admin**

**3.10.2.15 Tampilan Data Pemesanan Barang**

Table Pesan Barang

Filter Nama Produk    Keyword    Cari    Home

Id Member	Nama	Alamat	Nama Produk	No Handphone	Jumlah	Total	Tanggal	Aksi
								Hapus
								Hapus

**Gambar 3.29 Rancangan Tampilan Data Pemesanan Barang**

**3.10.2.16 Tampilan Data Pemesanan Jasa**

Table Pesan Jasa

Filter Nama Toko    Keyword    Cari    Home

Id Member	Nama	Alamat	Kelurahan	Kecamatan	No Handphone	Nama Toko	Jenis Layanan	Tanggal	Aksi
									Hapus
									Hapus

**Gambar 3.30 Rancangan Tampilan Data Pemesanan Jasa**

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Spesifikasi Perangkat Lunak dan Keras

##### a. Spesifikasi Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) yang digunakan dalam pembuatan aplikasi sebagai berikut:

- 1) Laptop *Processor Core i5*
- 2) *RAM 4 GB*
- 3) *Harddisk 1 terrabyte*
- 4) Printer standar

##### b. Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak (*software*), perangkat lunak yang dibutuhkan adalah:

- 1) Sistem operasi yang digunakan adalah *Microsoft Windows 10*.
- 2) Aplikasi program yang digunakan adalah *Android Studio* dan *sublime Text*.
- 3) Aplikasi rancangan yang digunakan adalah *Edraw Max* dan *Star Uml*.

##### c. Spesifikasi *Smartphone*

Perangkat *Smartphone* yang dibutuhkan adalah:

- 1) Layar *5.5 Inch 1280\*720 pixel*.
- 2) *Android 7.0 Nougat*.
- 3) *Memory RAM 3 GB*.



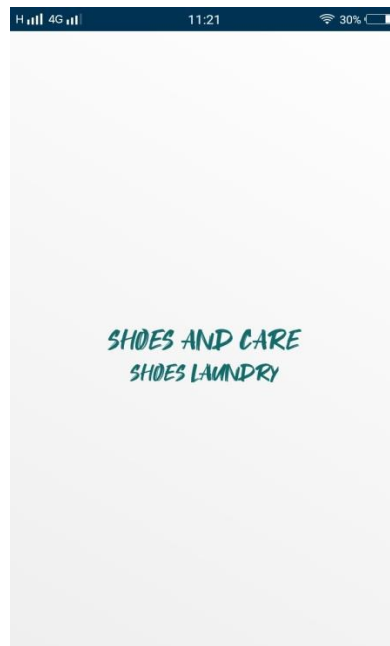
## 4.2 Implementasi

### 4.2.1 Hasil Pembuatan Aplikasi

Pada pembahasan hasil pembuatan aplikasi ini dijelaskan dalam bentuk tampilan program yang telah dijalankan (*running*). Adapun penjelasan fungsi atau kegunaan menu-menu yang terdapat pada tampilan aplikasi ini adalah sebagai berikut:

### 4.2.2 Tampilan Awal Aplikasi

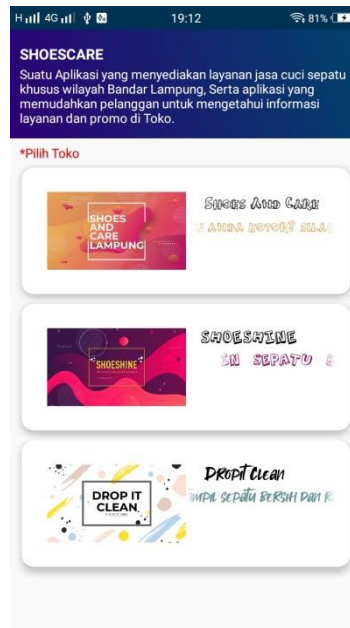
Pada tampilan awal aplikasi yaitu menampilkan *Splash Screen* aplikasi, dimana ketika *user* membuka aplikasi ini. seperti pada gambar 4.1 dibawah ini.



**Gambar 4.1 Tampilan Awal Aplikasi**

### 4.2.3 Tampilan Beranda Aplikasi

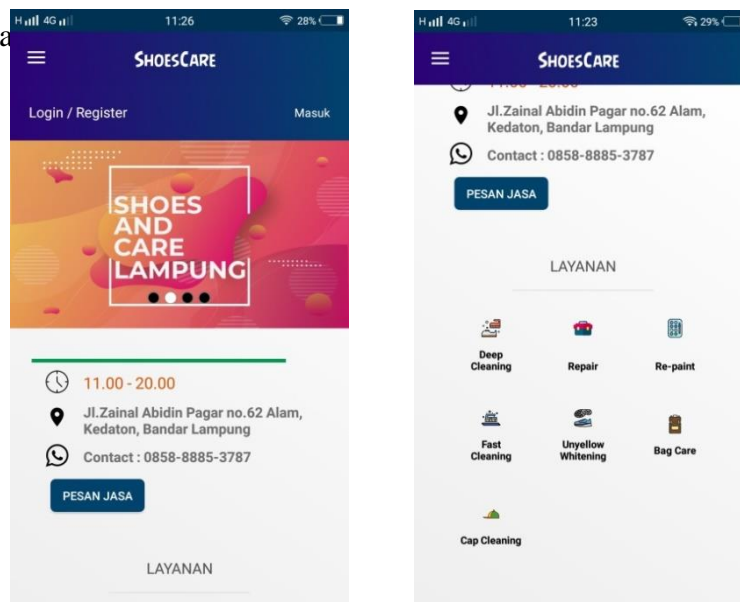
Pada tampilan Beranda aplikasi yaitu menampilkan Toko yang terdaftar di aplikasi, sehingga *user* dapat memilih toko. Seperti pada gambar 4.2 dibawah ini.



**Gambar 4.2 Tampilan Beranda Aplikasi**

#### 4.2.4 Tampilan Toko Shoes And Care

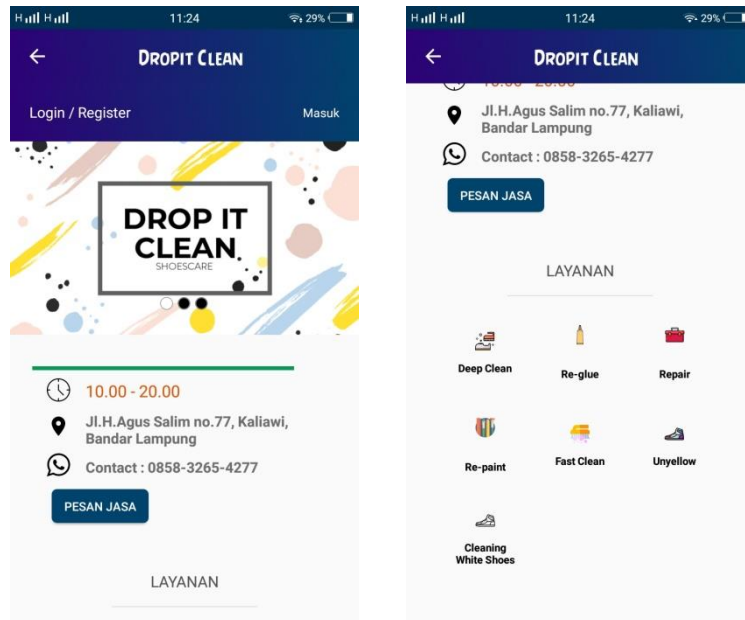
Pada tampilan toko Shoes And Care yaitu menampilkan deskripsi toko dan menu layanan yang tersedia, *user* dapat melihat menu layanan serta deskripsi dari masing-masing layanan. Seperti pada gambar 4.3 dibawah



**Gambar 4.3 Tampilan Toko Shoes And Care**

#### 4.2.5 Tampilan Toko Dropit Clean

Pada tampilan toko Dropit Clean yaitu menampilkan deskripsi toko dan menu layanan yang tersedia, *user* dapat melihat menu layanan serta deskripsi dari masing-masing layanan. Seperti pada gambar 4.4 dibawah ini :

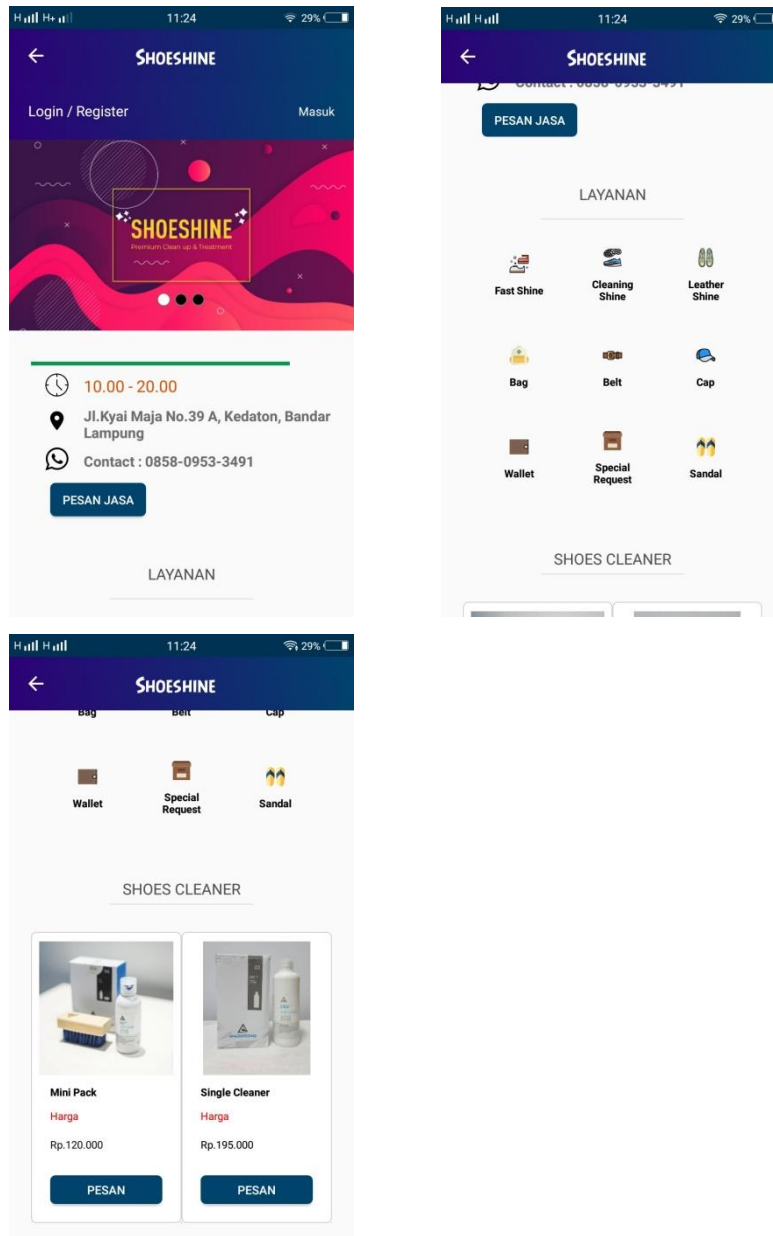


**Gambar 4.4 Tampilan Toko Dropit Clean**

#### **4.2.6 Tampilan Toko Shoeshine**

Pada tampilan toko Shoeshine yaitu menampilkan deskripsi toko dan menu layanan yang tersedia dan produk yang dijual, *user* dapat melihat menu layanan serta deskripsi dari masing-masing layanan. Seperti pada gambar 4.5 dibawah ini :

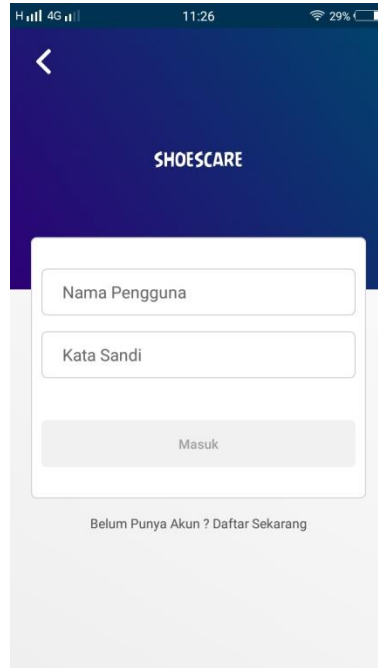




**Gambar 4.5 Tampilan Toko Shoeshine**

#### 4.2.7 Tampilan *Login*

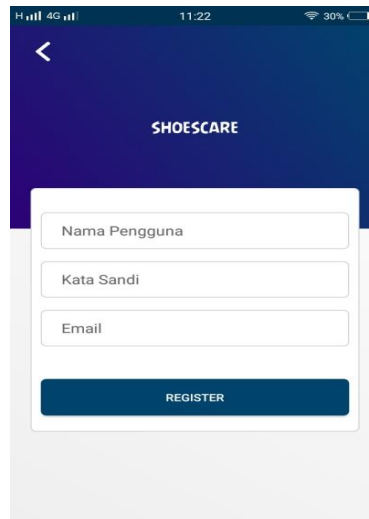
Pada tampilan *Login*, user dapat memasukkan username dan password yang telah di buat. Berikut gambar tampilan *Login* seperti pada gambar 4.6 dibawah ini:



**Gambar 4.6 Tampilan *Login***

#### **4.2.8 Tampilan *Register***

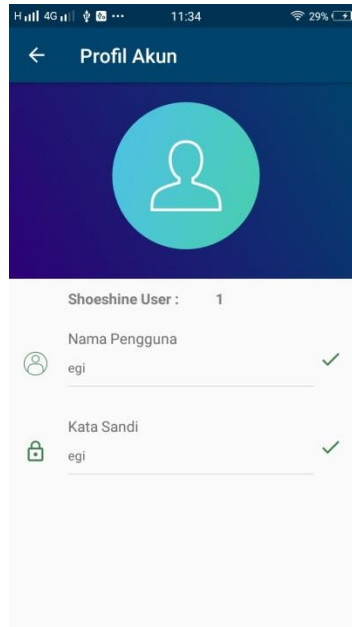
Pada tampilan *register*, *user* dapat melakukan registrasi dengan memasukan data diri beserta *username*, *password* dan *email*. Seperti yang terdapat pada gambar 4.7 dibawah ini.



**Gambar 4.7 Tampilan *Register***

#### **4.2.9 Tampilan Profil Akun**

Pada tampilan Profil Akun, *user* dapat melihat *username dan password* yang telah di buat di form registrasi. Berikut gambar tampilan profil akun seperti pada gambar 4.8 dibawah ini:



**Gambar 4.8 Tampilan Profil Akun**

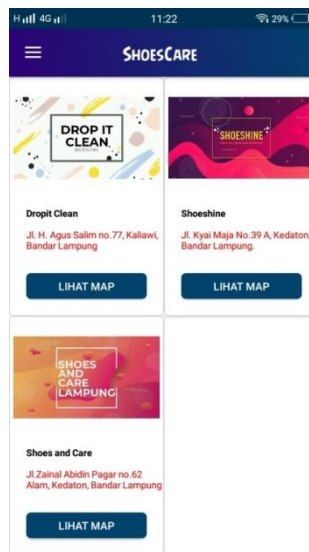
#### **4.2.10 Tampilan Pesan Jasa**

Pada tampilan Pesan Jasa, *user* dapat melakukan pemesanan jasa dengan mengisi form yang telah tersedia. Berikut gambar tampilan pesan jasa seperti pada gambar 4.9 dibawah ini:

## Gambar 4.9 Tampilan Pesan Jasa

### 4.2.11 Tampilan Menu Daftar Toko

Pada tampilan Menu Daftar Toko, *user* dapat melihat toko yang terdaftar . Berikut gambar tampilan menu daftar toko seperti pada gambar 4.10 dibawah ini:



Gambar 4.10 Tampilan Daftar Toko

### 4.2.12 Tampilan Lokasi Toko

Pada tampilan Lokasi Toko, *user* dapat melihat lokasi toko. Berikut gambar tampilan lokasi toko seperti pada gambar 4.11 dibawah ini:



**Gambar 4.11 Tampilan Lokasi Toko**

#### **4.2.13 Tampilan Pesan Barang**

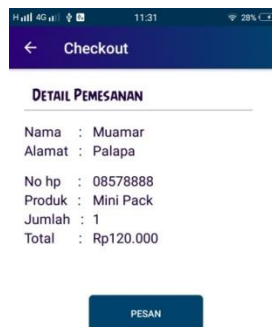
Pada tampilan Pesan Barang, *user* dapat melakukan pemesanan barang dengan mengisi form yang telah tersedia. Berikut gambar tampilan pesan barang seperti pada gambar 4.12 dibawah ini:

A screenshot of a mobile application's order form. The title bar is dark blue with a white back arrow and the text 'Pesan Barang'. The form consists of five input fields: 'Masukan Nama', 'Alamat', 'Telepon', 'Mini Pack', and 'Jumlah'. Below the form, the shipping method is indicated as 'Shipping : Cash on Delivery'. At the bottom, there is a dark blue button with the text 'PROSES'. The status bar at the top shows 4G LTE, signal strength, time 11:30, Wi-Fi, and 28% battery.

## Gambar 4.12 Tampilan Pesan Barang

### 4.2.14 Tampilan Checkout

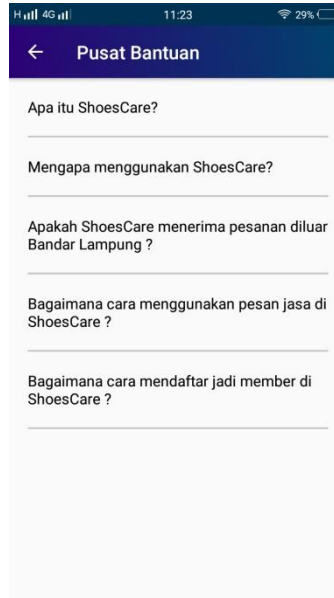
Pada tampilan *Checkout*, *user* dapat melihat detail pemesanan barang. Berikut gambar tampilan *Checkout* seperti pada gambar 4.13 dibawah ini:



## Gambar 4.13 Tampilan Checkout

### 4.2.15 Tampilan Menu Pusat Bantuan

Pada menu Pusat Bantuan, *user* dapat melihat informasi bantuan yang tersedia di aplikasi. Berikut gambar menu Pusat Bantuan seperti pada gambar 4.14 dibawah ini:



**Gambar 4.14 Tampilan menu Pusat Bantuan**

#### **4.2.16 Tampilan Menu Promosi**

Pada menu Promosi, *user* dapat melihat informasi promosi yang tersedia pada toko. Berikut gambar menu Promosi seperti pada gambar 4.15 dibawah ini:



**Gambar 4.15 Tampilan Menu Promosi**

#### 4.2.17 Tampilan Halaman Admin

Pada tampilan Halaman Admin. Berikut gambar halaman admin seperti pada gambar 4.16 dibawah ini:



Gambar 4.16 Tampilan Halaman Admin

#### 4.2.18 Tampilan Data Pemesanan Jasa

Pada tampilan Data Pemesanan Jasa. Admin dapat melihat Data pemesanan jasa Berikut gambar tampilan data pemesanan jasa seperti pada gambar 4.17 dibawah ini:

Tabel Pesan Jasa

Filter Nama Toko:  Keyword:  [Cari](#) [Home](#)

Id Member	Nama	Alamat	Kelurahan	Kecamatan	No Handphone	Nama Toko	Layanan	Jumlah	Total	Tanggal	Aksi	Hapus
1	egi	jalan umir hamzah perum indah	campang	campang raya	08978946025	Shoes and Care	Unyellow	1	Rp.....	2019-10-07	Cetak	Hapus
9	adit	jalan diponegoro gg pahi no 14	palapa	tanjung karang barat	0857582246977	Shoes and Care	Deep Cleaning	1	Rp.....	2019-10-07	Cetak	Hapus
8	rio	jalan waydadi gg cempaka no03	Sukarame	Sukarame	08127211586	Shoes and Care	Repaint	2	Rp.....	2019-10-07	Cetak	Hapus
7	roni	jalan maulana yusuf no 21	palapa	tannung karang pusat	085758221554	Shoes and Care	Deep Cleaning	1	Rp.....	2019-10-06	Cetak	Hapus

Gambar 4.17 Tampilan Data Pemesanan Jasa

#### 4.2.19 Tampilan Data Pemesanan Barang



Pada tampilan Data Pemesanan Barang. Admin dapat melihat data pemesanan barang Berikut gambar tampilan data pemesanan barang seperti pada gambar 4.18 dibawah ini:

Table Pesan Barang

Filter Nama Produk  Keyword

Id Member	Nama	Alamat	Kecamatan	Nama Produk	Nohp	Jumlah	Total	Tanggal	Aksi	Hapus
8	nopal	jln way ratei gg ubi	gedong air	Mini Pack	085758221226	1	Rp120.000	2019-09-19	<input type="button" value="Cetak"/>	<input type="button" value="Hapus"/>
7	doni	jln hayam wuruk gg damay	sukarame	Mini Pack	08127211854	1	Rp120.000	2019-09-14	<input type="button" value="Cetak"/>	<input type="button" value="Hapus"/>
6	raditya	jalan wonosibo gg pahi no2	sukarame	Single Cleaner	0857845121379	2	Rp390.000	2019-09-10	<input type="button" value="Cetak"/>	<input type="button" value="Hapus"/>
5	muamar	jalan dlponegoro no 38	wayhalim	Single Cleaner	085758221228	1	Rp195.000	2019-09-10	<input type="button" value="Cetak"/>	<input type="button" value="Hapus"/>

**Gambar 4.18 Tampilan Data Pemesanan Barang**

### 4.3 Kelayakan Sistem

Berikut merupakan hasil kelayakan sistem aplikasi Shoes And Care yang telah dibuat dengan menggunakan tabel pengujian dibawah ini.

**Tabel 4.1 Kelayakan Sistem**

No	Keterangan	Bisa	Tidak
	Membuka Aplikasi ShoesCare	√	
	Registrasi User	√	
	Login User	√	
	Beranda	√	
	Menu Aplikasi: Home Daftar Toko Promosi Pusat Bantuan Tentang Aplikasi	√	
	Opikasi Toko	√	
	Pesanan Jasa	√	
	Pesanan Barang	√	
	Profil Akun	√	
	Halaman Admin	√	
10.	Menu Data Pemesanan Jasa	√	
11.	Menu Data Pemesanan Barang	√	

## **BAB V**

### **SIMPULAN DAN SARAN**

Berdasarkan penelitian analisis dan pembahasan pada bab sebelumnya maka dapat di ambil beberapa simpulan mengenai Sistem Informasi Jasa *Customer Service* Pada Shoes And Care Lampung Berbasis Android sebagai berikut.

#### **5.1 Simpulan**

Berdasarkan hasil dan pembahasan, simpulan dari penelitian ini sebagai berikut.

1. Dengan adanya sistem informasi jasa *customer service* pada shoes and care lampung berbasis android, dapat mempermudah pelanggan untuk melakukan pemesanan jasa cuci sepatu dan dapat mengetahui informasi menu layanan serta promosi yang disediakan oleh toko.
2. Dengan adanya aplikasi dapat membantu pihak toko untuk memfilterkan pengolahan data pemesanan dan mengetahui member yang ada pada masing-masing toko.
3. Dengan adanya aplikasi ini di harapkan juga dapat meningkatkan pelayanan pada toko yang telah terdaftar.

#### **5.2 Saran**

Saran yang dapat diberikan pada penelitian ini yaitu :

1. Pada aplikasi ini kurangnya sistem pembayaran *online* via ATM, Bagi pengembang selanjutnya diharapkan dapat menambahkan sistem pembayaran via ATM pada aplikasi ini.
2. Pada aplikasi ini kurangnya sistem penghitungan ongkos sesuai jarak penjemputan, Bagi pengembang selanjutnya diharapkan dapat menambahkan sistem penghitungan ongkos sesuai jarak pada aplikasi ini.

3. Penelitian selanjutnya diharapkan untuk mengembangkan aplikasi ini dalam penilaian rating pada masing-masing toko.

## DAFTAR PUSTAKA

Febriani, O. M., & Permadi, A. B. (2017). Implementasi Sistem Aplikasi Data Bimbingan dan Pelanggaran Siswa pada Sekolah Menengah Atas di Lampung Tengah dengan Metode Analisis dan Desain Sistem Terdistribusi (SSAD). *EXPERT*, 7(1).

Indarto, 2013. Sistem Informasi Geografis, Penerbit Graha Ilmi, Yogyakarta.

Indera, 2016. Rancangan Bangun Aplikasi Visualisasi Kamus Bahasa Lampung Berbasis Android, Penerbit Universitas Bandar Lampung, Bandar Lampung.

Jubilee Enterprise, 2018. HTML, PHP, dan MySQL Untuk Pemula, Penerbit PT Elex Media Komputindo, Yogyakarta.

Machtal Nuar Selwa Perdana. 2017. Pembuatan Aplikasi Pemesanan Jasa Foto Dokumentasi Dan Video Shooting Pada Studio Citra Berbasis Android . Yogyakarta: Informatika Universitas AMIKOM Yogyakarta.

Muhammad Fadhli Amrulloh. 2018. Pembuatan Aplikasi "Shoes Care" Berbasis Sistem Android . Yogyakarta: Universitas AMIKOM Yogyakarta.

Nadia Firly. 2018. Create Your Own Android Application. Penerbit PT Elex Media Komputindo, Bogor.

Nurita Evitarina. 2016. Rancang Bangun Aplikasi Pemesanan Barang Berbasis Android Pada Mini Market Faras Pangkalpinang . Pangkalpinang: STMIK ATMA LUHUR PANGKALPINANG.

Orien Rindy Erika. 2016. Aplikasi Pencarian Letak ATM Berbasis Android Dengan GIS Di Kota Bandar Lampung . Bandar Lampung: Unila.

Rahayu Tri Utami. 2014. *Pengaruh Customer Service Terhadap Loyalitas Nasabah Di PT. AIA Financial Jakarta*. Jakarta.

Rosa A.S M. Shalahuddin. 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Penerbit Informatika, Bandung.

Tata Sutabri. 2014. *Analisis Sistem Informasi*. Penerbit Andi, Yogyakarta.

Yenita Dwi Setiyawati. 2016. *Pembuatan Aplikasi Antar-Jemput Laundry Berbasis Web Service pada Platform Android*. Semarang: Universitas Diponegoro.

Yoko Andramawan. 2018. *Rancang Bangun Aplikasi Pemesanan Jasa Perbaikan Komputer, Laptop, dan Smartphone Berbasis Android* . Medan: Universitas Potensi Utama.

# LAMPIRAN

Program Android

## BERANDA

```
package com.amar.shoes.shoescare;

import android.content.Intent;
import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class SplashScreen2 extends AppCompatActivity {
    TextView text1, text2, text3, text4, text5, text6;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen2);

        text1 = (TextView)
findViewById(R.id.marque_scrolling_text);
        Typeface customfont =
Typeface.createFromAsset(getAssets(), "fonts/burnstown.ttf");
        text1.setTypeface(customfont);
        text1.setTextSize(20);

        text2 = (TextView) findViewById(R.id.scl);
        Typeface customfonts =
Typeface.createFromAsset(getAssets(), "fonts/burnstown.ttf");
        text2.setTypeface(customfonts);
        text2.setTextSize(21);

        text3 = (TextView) findViewById(R.id.marque2);
        Typeface customfontt =
Typeface.createFromAsset(getAssets(), "fonts/actionj.ttf");
        text3.setTypeface(customfontt);
        text3.setTextSize(18);

        text4 = (TextView) findViewById(R.id.ss1);
        Typeface customfontss =
Typeface.createFromAsset(getAssets(), "fonts/actionj.ttf");
        text4.setTypeface(customfontss);
        text4.setTextSize(20);

        text5 = (TextView) findViewById(R.id.marque3);
        Typeface customfontst =
Typeface.createFromAsset(getAssets(), "fonts/Takota.ttf");
        text5.setTypeface(customfontst);
        text5.setTextSize(18);

        text6 = (TextView) findViewById(R.id.dcl);
        Typeface customfontsss =
Typeface.createFromAsset(getAssets(), "fonts/Takota.ttf");
        text6.setTypeface(customfontsss);
        text6.setTextSize(20);

        TextView marque = (TextView)
this.findViewById(R.id.marque_scrolling_text);
        marque.setSelected(true);
    }
}
```



```

        TextView marquee = (TextView)
this.findViewById(R.id.marque2);
        marquee.setSelected(true);

        TextView marquee3 = (TextView)
this.findViewById(R.id.marque3);
        marquee3.setSelected(true);

        getSupportActionBar().setDisplayShowTitleEnabled(false);
        getSupportActionBar().hide();
    }

    public void sac(View view) {
        Intent intent = new Intent(this, Home.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.putExtra("EXIT", true);
        startActivity(intent);
        finish();
    }

    public void ss(View view) {
        Intent intent = new Intent(this, Shoehine.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.putExtra("EXIT", true);
        startActivity(intent);
        finish();
    }

    public void dc(View view) {
        Intent intent = new Intent(this, Dropitclean.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.putExtra("EXIT", true);
        startActivity(intent);
        finish();
    }
}

```

## TOKO SHOES AND CARE

```

package com.amar.shoes.shoescare;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomSheetDialog;
import android.support.design.widget.NavigationView;

```

```

import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.text.Html;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.synnapps.carouselview.CarouselView;
import com.synnapps.carouselview.ImageListener;

public class Home extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    ActionBarDrawerToggle toggle;
    DrawerLayout drawerLayout;

    SharedPreferences sharedPreferences;

    private BottomSheetDialog bottomSheetDialog;

    CarouselView caresoul;
    int[] imgae = {R.mipmap.shoesandcare1, R.mipmap.shoesandcare,
R.mipmap.shoesandcare, R.mipmap.shoesandcare};

    Button jasaa;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        jasaa = (Button) findViewById(R.id.pindahjasa);

        jasaa.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDialog();
            }
        });

        caresoul = (CarouselView) findViewById(R.id.viewflip);
        caresoul.setPageCount(imgae.length);
        caresoul.setImageListener(imad);

        sharedPreferences =
getSharedPreferences(LoginnActivity.Shared_Name,
Context.MODE_PRIVATE);

```

```

getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);

LayoutInflater inflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
View v = inflater.inflate(R.layout.toolbars, null);

ActionBar.LayoutParams p = new ActionBar.LayoutParams(
    ViewGroup.LayoutParams.MATCH_PARENT,
    ViewGroup.LayoutParams.MATCH_PARENT,
    Gravity.CENTER);

TextView textV = (TextView)v.findViewById(R.id.textapp);
textV.setText("ShoesCare");
Typeface customfont =
Typeface.createFromAsset(getAssets(), "fonts/angrybirds.ttf");
textV.setTypeface(customfont);
textV.setTextSize(23);
textV.setGravity(Gravity.CENTER);

getSupportActionBar().setCustomView(v, p);
getSupportActionBar().setDisplayShowTitleEnabled(true);

getSupportActionBar().setBackgroundDrawable(getResources().getDrawable(R.drawable.linearbackground));
getSupportActionBar().setElevation(0);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
drawerLayout = (DrawerLayout)
findViewById(R.id.menu_utama);
toggle = new ActionBarDrawerToggle(this, drawerLayout,
R.string.open, R.string.close);
toggle.syncState();
drawerLayout.addDrawerListener(toggle);

NavigationView navigationView = (NavigationView)
findViewById(R.id.navigation);
navigationView.setNavigationItemSelectedListener(this);

}

private void showDialog() {

    AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
        this);

    alertDialogBuilder
        .setMessage(Html.fromHtml("<font
color='#000'>Silahkan Login Terlebih Dahulu</font>"))
        .setIcon(R.mipmap.sulitttcare)
        .setCancelable(false)
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {

```

```

        public void onClick(DialogInterface dialog, int
id) {
            Intent i = new Intent (Home.this,
LoginnActivity.class);
            startActivity(i);
            finish();
        }
    })
    .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,
int id) {
            // jika tombol ini diklik, akan menutup
dialog
            // dan tidak terjadi apa2
            dialog.cancel();
        }
    });

    // membuat alert dialog dari builder
    AlertDialog alertDialog = alertDialogBuilder.create();

    // menampilkan alert dialog
    alertDialog.show();
}

    ImageListener imad = new ImageListener() {
        @Override
        public void setImageForPosition(int position, ImageView
imageView) {
            imageView.setImageResource(imgae[position]);
        }
    };

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (toggle.onOptionsItemSelected(item)) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem
item) {
        switch (item.getItemId()){

            case R.id.home :
                break;
            case R.id.toko :
                Intent intent = new Intent(Home.this,
DaftarToko.class);
                startActivity(intent);
                overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
                break;
            case R.id.beranda :
                Intent intento = new Intent(Home.this,

```

```

SplashScreen2.class);
        startActivity(intent);
        overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
        break;
        case R.id.infoapp:
            Intent intentss = new Intent(Home.this,
InfoAplikasi.class);
            startActivity(intentss);
            overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
            break;
        case R.id.Tentangapp:
            Intent intentssa = new Intent(Home.this,
TentangAplikasi.class);
            startActivity(intentssa);
            overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
            break;
        case R.id.promosi:
            Intent intentssas = new Intent(Home.this,
Coba.class);
            startActivity(intentssas);
            overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
            break;
    }
    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.menu_utama);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    }
    return false;
}

public void login(View view) {
    Intent intent = new Intent(Home.this,
LoginnActivity.class);
    startActivity(intent);
    overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
}

@Override
public void onBackPressed() {
    Intent a = new Intent(Intent.ACTION_MAIN);
    a.addCategory(Intent.CATEGORY_HOME);
    a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(a);
}

public void deepclean(View view) {
    final View bottomsheetslayout =
getLayoutInflater().inflate(R.layout.deep_cleaning, null);

```

```

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void repair(View view) {
        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.repair, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void repaint(View view) {
        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.repaint, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void fastcleaning(View view) {
        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.fast_cleaning, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void unyellow(View view) {
        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.unyellow, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void bag(View view) {
        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.bagcare, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }

    public void cap(View view) {

```

```

        final View bottomsheetlayout =
getLayoutInflater().inflate(R.layout.cap_cleaning, null);

        bottomSheetDialog = new BottomSheetDialog(this,
R.style.BottomSheets);
        bottomSheetDialog.setContentView(bottomsheetlayout);
        bottomSheetDialog.show();
    }
}

```

## PESAN JASA

```

package com.amar.shoes.shoescare;

import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.amar.shoes.shoescare.pesan.MySQLClient;
import com.amar.shoes.shoescare.pesan.Spacecraft;

public class PesanjasaSC extends AppCompatActivity {

    public static final String KEY_USERNAME="username";
    public static final String KEY_ID="Id_username";

    String user,memberr ;

    private EditText name, alamat, kelurahan, kecamatan, nohp,
member;
    private Spinner spinner1, spinner2, spinner3;
    private Button btncsave;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pesanjasa_sc);

        btncsave = (Button) findViewById(R.id.btn_pesan);

        name= (EditText) findViewById(R.id.txtnama);
        name.setSelected(false);
        user = getIntent().getStringExtra(KEY_USERNAME);
        name.setText(user);

        member=(EditText) findViewById(R.id.idmember);
        member.setSelected(false);
        memberr = getIntent().getStringExtra(KEY_ID);
        member.setText(memberr);
    }
}

```

```

        this.initializeViews();

        populatePropellants();
        populatePropellants2();
        populatePropellants3();

        this.handleClickEvents();

        TextView textV = (TextView) findViewById(R.id.nameapp1);
        textV.setText("Shoes Laundry");
        Typeface customfont =
Typeface.createFromAsset(getAssets(), "fonts/angrybirds.ttf");
        textV.setTypeface(customfont);
        textV.setTextSize(23);

        getSupportActionBar().setDisplayShowTitleEnabled(false);
        getSupportActionBar().hide();

    }

    private void populatePropellants3() {

        ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown
_item);

        adapter.add("1");
        adapter.add("2");
        adapter.add("3");
        adapter.add("4");
        adapter.add("5");

        spinner3.setAdapter(adapter);
        spinner3.setSelection(2);
    }

    private void populatePropellants2() {

        ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown
_item);

        adapter.add("Deep Cleaning");
        adapter.add("Repair");
        adapter.add("Repaint");
        adapter.add("Unyellow");
        adapter.add("Bag Care");
        adapter.add("Cap Cleaning");
        adapter.add("Fast cleaning");

        spinner2.setAdapter(adapter);
    }

```



```

        spinner2.setSelection(1);
    }

    private void handleClickEvents() {

        btnsave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String mem = member.getText().toString();
                String na = name.getText().toString();
                String al = alamat.getText().toString();
                String kel = kelurahan.getText().toString();
                String kec = kecamatan.getText().toString();
                String no = nohp.getText().toString();
                String natok =
spinner1.getSelectedItem().toString();
                String jelay =
spinner2.getSelectedItem().toString();
                String jubar =
spinner3.getSelectedItem().toString();
                if ((na.length() < 1 || al.length() < 1 ||
kel.length() < 1 || kec.length() < 1 || kec.length() < 1 ||
no.length() < 1 || natok.length() < 1 || jelay.length() < 1 ||
jubar.length() < 1 )) {
                    Toast.makeText(PesanjasaSC.this, "Please Enter
all Fields", Toast.LENGTH_SHORT).show();
                } else {
                    Spacecraft s = new Spacecraft();
                    s.setId_member(mem);
                    s.setNama(na);
                    s.setAlamat(al);
                    s.setKelurahan(kel);
                    s.setKecamatan(kec);
                    s.setNohp(no);
                    s.setNamatoko(natok);
                    s.setJenis(jelay);
                    s.setJumlahh(jubar);

                    new
MySQLClient(PesanjasaSC.this).add(s, member, name, alamat,
kelurahan, kecamatan, nohp, spinner1, spinner2, spinner3);
                }
            }
        });
    }
}

```

```

        private void populatePropellants() {

            ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown
_item);

            adapter.add("Shoes and Care");

            spinner1.setAdapter(adapter);
            spinner1.setSelection(0);

        }

        private void initializeViews() {

            member = (EditText) findViewById(R.id.idmember);
            name= (EditText) findViewById(R.id.txtnama);
            alamat= (EditText) findViewById(R.id.txtalamat);
            kelurahan= (EditText) findViewById(R.id.txtkelurahan);
            kecamatan= (EditText) findViewById(R.id.txtkecamatan);
            nohp= (EditText) findViewById(R.id.txtnohp);
            spinner1= (Spinner) findViewById(R.id.spinner1);
            spinner2= (Spinner) findViewById(R.id.spinner2);
            spinner3= (Spinner) findViewById(R.id.spinner3);
            btnsave= (Button) findViewById(R.id.btn_pesan);

        }

        @Override
        public void onBackPressed() {
            Intent intent = new Intent(this, LoginnActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.putExtra("EXIT", true);
            startActivity(intent);
            overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
            finish();
        }
    }
}

```

## **PESAN JASA (KONEKSI)**

```

package com.amar.shoes.shoescare.pesan;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.support.v7.app.NotificationCompat;
import android.view.View;
import android.widget.EditText;

```

```

import android.widget.Spinner;
import android.widget.Toast;

import com.amar.shoes.shoescare.Home;
import com.amar.shoes.shoescare.R;
import com.androidnetworking.AndroidNetworking;
import com.androidnetworking.error.ANError;
import com.androidnetworking.interfaces.JSONArrayRequestListener;

import org.json.JSONArray;
import org.json.JSONException;

public class MySQLClient {

    private static final String
DATA_INSERT_URL="http://u312737575.hostingerapp.com/android/CRUD.php";

    private final Context c;

    public MySQLClient(Context c) {
        this.c = c;
    }

    public void add(Spacecraft s, final View...inputViews)
    {
        final ProgressDialog dialog = new ProgressDialog(c);
        dialog.setMessage("Sedang Memuat");
        dialog.show();

        if(s==null)
        {
            Toast.makeText(c, "No Data To Save",
Toast.LENGTH_SHORT).show();
            dialog.dismiss();
        }
        else
        {
            AndroidNetworking.post(DATA_INSERT_URL)
                .addBodyParameter("action","save")

.addBodyParameter("id_member",s.getId_member())
                .addBodyParameter("nama",s.getNama())
                .addBodyParameter("alamat",s.getAlamat())

.addBodyParameter("kelurahan",s.getKelurahan())

.addBodyParameter("kecamatan",s.getKecamatan())
                .addBodyParameter("nohp",s.getNohp())
                .addBodyParameter("namatoko",s.getNamatoko())
                .addBodyParameter("layanan",s.getJenis())

```

```

        .addBodyParameter("jumlah", s.getJumlahh())
        .setTag("TAG_ADD")
        .build()
        .getAsJSONArray(new JSONArrayRequestListener()
    {
        @Override
        public void onResponse(JSONArray response)
    {
        if(response != null)
            try {
                String responseString =
response.get(0).toString();
                Toast.makeText(c, "PHP SERVER
RESPONSE : " + responseString, Toast.LENGTH_SHORT).show();
                dialog.dismiss();
                Intent intent = new Intent(c,
Home.class);

                c.startActivity(intent);

                if
(responseString.equalsIgnoreCase("Success")) {
                    EditText member =
                    EditText name = (EditText)
                    EditText alamat =
                    EditText kelurahan =
                    EditText kecamatan =
                    EditText nohp = (EditText)
                    Spinner spinner1 =
                    Spinner spinner2 =
                    Spinner spinner3 =

                    member.setText("");
                    name.setText("");
                    alamat.setText("");
                    kelurahan.setText("");
                    kecamatan.setText("");
                    nohp.setText("");
                    spinner1.setSelection(0);
                    spinner2.setSelection(1);
                    spinner3.setSelection(2);

                notifTempllate(tittle,message);

            }else
            {
                Toast.makeText(c, "PHP

```

```

WASN'T SUCCESSFUL. ", Toast.LENGTH_SHORT).show();
                                dialog.dismiss();

                                }

                                } catch (JSONException e) {
                                    e.printStackTrace();
                                    Toast.makeText(c, "GOOD
RESPONSE BUT JAVA CAN'T PARSE JSON IT RECEIVED : "+e.getMessage(),
Toast.LENGTH_SHORT).show();

                                    dialog.dismiss();
                                }

                                }

                                @Override
                                public void onError(ANError anError) {
                                    Toast.makeText(c, "UNSUCCESSFUL :
ERROR IS : "+anError.getMessage(), Toast.LENGTH_SHORT).show();
                                }

                                });

                                }
                                }

                                private void notifTempllate(String tittle, String message) {
                                    final Intent intent = new Intent(c, Home.class);
                                    PendingIntent pendingIntent =
PendingIntent.getActivity(c, 15, intent,
PendingIntent.FLAG_UPDATE_CURRENT);

                                    NotificationCompat.Builder mbuilder =
(NotificationCompat.Builder) new NotificationCompat.Builder(c)
                                        .setSmallIcon(R.mipmap.sulitttcare)
                                        .setContentTitle(tittle)
                                        .setContentText(message)
                                        .setContentIntent(pendingIntent)
                                        .setAutoCancel (true)
                                        .setDefaults(Notification.DEFAULT_LIGHTS |
Notification.DEFAULT_SOUND | Notification.FLAG_AUTO_CANCEL);
                                    NotificationManager notificationmanager =
(NotificationManager)
c.getSystemService(Context.NOTIFICATION_SERVICE);
                                    notificationmanager.notify(1,mbuilder.build());
                                }

                                final String message = "Terima Kasih, Tunggu Konfirmasi dari
Toko.";
                                final String tittle = "Verivikasi Pemesanan";
                                }
}

LOKASI (PROSES)
package com.amar.shoes.shoescare.toko;

import android.content.Intent;

```

```

import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.amar.shoes.shoescare.DaftarToko;
import com.amar.shoes.shoescare.Dropitclean;
import com.amar.shoes.shoescare.Home;
import com.amar.shoes.shoescare.Map;
import com.amar.shoes.shoescare.MapDrop;
import com.amar.shoes.shoescare.MapShine;
import com.amar.shoes.shoescare.Promosi;
import com.amar.shoes.shoescare.R;
import com.amar.shoes.shoescare.Shoehine;
import com.bumptech.glide.Glide;
import com.bumptech.glide.load.resource.drawable.GlideDrawable;
import com.bumptech.glide.request.RequestListener;
import com.bumptech.glide.request.target.Target;

import java.util.List;

public class AdapterList extends
RecyclerView.Adapter<AdapterList.ViewHolder> {

    private List<ModelList> modellists;
    private DaftarToko context;

    public AdapterList(List<ModelList> modellists, DaftarToko
context) {
        this.modellists = modellists;
        this.context = context;
    }

    @Override
    public AdapterList.ViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        View layout =
LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_r
ow, parent, false);
        ViewHolder holder = new ViewHolder(layout);
        return holder;
    }

    @Override
    public void onBindViewHolder(final AdapterList.ViewHolder
holder, int position) {
        final ModelList list = modellists.get(position);

        holder.nama_toko1.setText(list.getNama_toko1());
        holder.nama_toko2.setText(list.getNama_toko2());
        holder.nama_toko3.setText(list.getNama_toko3());
        holder.alamat_1.setText(list.getAlamat_toko1());
    }
}

```

```

holder.alamat_2.setText(list.getAlamat_toko2());
holder.alamat_3.setText(list.getAlamat_toko3());

holder.mep.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context,
MapDrop.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

holder.mep2.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context,
MapShine.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

holder.mep3.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context, Map.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

holder.tokol.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context,
Dropitclean.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

holder.toko2.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {

```

```

        Intent intent = new Intent(context,
Shoehine.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

holder.toko3.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context, Home.class);

        context.startActivity(intent);
        context.overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
});

Glide.with(context).load(list.getGambar1()).listener(new
RequestListener<String, GlideDrawable>() {
    @Override
    public boolean onException(Exception e, String model,
Target<GlideDrawable> target, boolean isFirstResource) {
        Toast.makeText(context, "Gagal memuat gambar",
Toast.LENGTH_SHORT).show();
        return false;
    }

    @Override
    public boolean onResourceReady(GlideDrawable resource,
String model, Target<GlideDrawable> target, boolean
isFromMemoryCache, boolean isFirstResource) {
        holder.imageView6.setVisibility(View.GONE);
        holder.imageView1.setVisibility(View.VISIBLE);
        return false;
    }
}).crossFade()
    .into(holder.imageView1);

Glide.with(context).load(list.getGambar2()).listener(new
RequestListener<String, GlideDrawable>() {
    @Override
    public boolean onException(Exception e, String model,
Target<GlideDrawable> target, boolean isFirstResource) {
        Toast.makeText(context, "Gagal memuat gambar",
Toast.LENGTH_SHORT).show();
        return false;
    }

    @Override
    public boolean onResourceReady(GlideDrawable resource,
String model, Target<GlideDrawable> target, boolean

```



```

isFromMemoryCache, boolean isFirstResource) {
    holder.imageView4.setVisibility(View.GONE);
    holder.imageView2.setVisibility(View.VISIBLE);
    return false;
}
}).crossFade()
    .into(holder.imageView2);
Glide.with(context).load(list.getGambar3()).listener(new
RequestListener<String, GlideDrawable>() {
    @Override
    public boolean onException(Exception e, String model,
Target<GlideDrawable> target, boolean isFirstResource) {
        Toast.makeText(context, "Gagal memuat gambar",
Toast.LENGTH_SHORT).show();
        return false;
    }

    @Override
    public boolean onResourceReady(GlideDrawable resource,
String model, Target<GlideDrawable> target, boolean
isFromMemoryCache, boolean isFirstResource) {
        holder.imageView5.setVisibility(View.GONE);
        holder.imageView3.setVisibility(View.VISIBLE);
        return false;
    }
}).crossFade()
    .into(holder.imageView3);
}

@Override
public int getItemCount() {
    return modelLists.size();
}

public class HolderItem extends RecyclerView.ViewHolder {

    ImageView imageView1, imageView2, imageView3, imageView4,
imageView5, imageView6;
    CardView toko1, toko2, toko3;
    TextView nama_toko1, nama_toko2, nama_toko3, alamat_1,
alamat_2, alamat_3, promol, promo2;
    Button mep, mep2, mep3;

    public HolderItem(View itemView) {
        super(itemView);

        imageView1 =
(ImageView) itemView.findViewById(R.id.img_cover1);
        imageView2 =
(ImageView) itemView.findViewById(R.id.img_cover2);
        imageView3 =
(ImageView) itemView.findViewById(R.id.img_cover3);
        nama_toko1 =
(TextView) itemView.findViewById(R.id.namatoko1);
        nama_toko2 =

```

```

        (TextView) itemView.findViewById(R.id.namatoko2);
        nama_toko3 =
        (TextView) itemView.findViewById(R.id.namatoko3);
        alamat_1 = (TextView)
        itemView.findViewById(R.id.alamat1);
        alamat_2 =
        (TextView) itemView.findViewById(R.id.alamat2);
        alamat_3 =
        (TextView) itemView.findViewById(R.id.alamat3);
        mep = (Button) itemView.findViewById(R.id.maplihat);
        mep2 = (Button) itemView.findViewById(R.id.maplihat2);
        mep3 = (Button) itemView.findViewById(R.id.maplihat3);
        imageView6 =
        (ImageView) itemView.findViewById(R.id.loadgamb);
        imageView4 =
        (ImageView) itemView.findViewById(R.id.loadgamb1);
        imageView5 =
        (ImageView) itemView.findViewById(R.id.loadgamb3);
        toko1 = (CardView) itemView.findViewById(R.id.dropit);
        toko2 = (CardView) itemView.findViewById(R.id.shine);
        toko3 =
        (CardView) itemView.findViewById(R.id.shoescare);

    }
}
}

```

## LOKASI (KONEKSI)

```

package com.amar.shoes.shoescare;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.MenuItem;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;

```

```

import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class Map extends AppCompatActivity implements
    OnMapReadyCallback,
    GoogleMap.OnMyLocationButtonClickListener{

    MapFragment mapFragment;
    GoogleMap gMap;
    MarkerOptions markerOptions = new MarkerOptions();
    CameraPosition cameraPosition;
    LatLng center, latLng;
    String title;

    GoogleApiClient mgoogleapi;
    private LocationRequest locationRequest;
    private Location mlocation;

    public static final String ID = "id_lokasi";
    public static final String TITLE = "nama_lokasi";
    public static final String LAT = "lat";
    public static final String LNG = "lng";

    private String url =
"http://u312737575.hostingerapp.com/android/markermap.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);

        getSupportActionBar().setTitle("Lokasi");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getSupportActionBar().setBackgroundDrawable(getResources().getDrawable(R.drawable.linearbackground));

        mapFragment =
        (MapFragment) getFragmentManager().findFragmentById(R.id.maps);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        gMap = googleMap;

        center = new LatLng(-5.4021256, 105.2612953);
        cameraPosition = new
CameraPosition.Builder().target(center).zoom(10).build();

        googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));

```

```

markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.s
toee));

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 1);

            } else {
                gMap.setMyLocationEnabled(true);
            }
        } else {
            gMap.setMyLocationEnabled(true);
        }

        gMap.setOnMyLocationButtonClickListener(this);

        getMarkers();
    }

    private void getMarkers() {
        StringRequest str = new StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.e("Response: ", response.toString());

                try {
                    JSONObject jsonObj = new JSONObject(response);
                    String getobjrct =
jsonObj.getString("lokasitoko");
                    JSONArray jsonarray = new
JSONArray(getobjrct);

                    for (int i = 0; i < jsonarray.length(); i++) {
                        JSONObject jsonobject =
jsonarray.getJSONObject(i);
                        title = jsonobject.getString(TITLE);
                        latLng = new
LatLng(Double.parseDouble(jsonobject.getString(LAT)),
Double.parseDouble(jsonobject.getString(LNG)));

                        addMarker(latLng, title);
                    }

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        })
    }
}

```

```

    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error)
    {
        Log.e("Error: ", error.getMessage());
        Toast.makeText(Map.this,
error.getMessage(), Toast.LENGTH_LONG).show();
        }
    });
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    requestQueue.add(str);
}

private void addMarker(LatLng latLng, final String title) {
    markerOptions.position(latLng);
    markerOptions.title(title);
    gMap.addMarker(markerOptions);

    gMap.setOnInfoWindowClickListener(new
GoogleMap.OnInfoWindowClickListener() {
        @Override
        public void onInfoWindowClick(Marker marker) {
            Toast.makeText(getApplicationContext(),
marker.getTitle(), Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
public boolean onMyLocationButtonClicked() {
    Toast.makeText(this, "My Location",
Toast.LENGTH_SHORT).show();
    return true;
}

@Override
public void onBackPressed() {
    Intent intent = new Intent(this, DaftarToko.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.putExtra("EXIT", true);
    startActivity(intent);
    overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    finish();
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home :
            Intent intent = new Intent(this,
DaftarToko.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.putExtra("EXIT", true);

```

```

        startActivity(intent);
        overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
        finish();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
}

```

## PROMOSI

```

package com.amar.shoes.shoescare;

import android.content.Intent;
import android.net.Uri;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MenuItem;

import com.amar.shoes.shoescare.Fragment.PagerAdapter;
import com.amar.shoes.shoescare.Fragment.Tab1;
import com.amar.shoes.shoescare.Fragment.Tab2;
import com.amar.shoes.shoescare.Fragment.Tab3;
import com.amar.shoes.shoescare.Fragment.Tab4;

public class Coba extends AppCompatActivity implements
Tab1.OnFragmentInteractionListener,
Tab2.OnFragmentInteractionListener,
Tab3.OnFragmentInteractionListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_coba);

        getSupportActionBar().setTitle("Promosi");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getSupportActionBar().setBackgroundDrawable(getResources().getDrawable(R.drawable.linearbackground));

        TabLayout tabLayout = (TabLayout)
findViewById(R.id.tablayout);
        tabLayout.addTab(tabLayout.newTab().setText("Shoes and
Care"));
        tabLayout.addTab(tabLayout.newTab().setText("Shoeshine"));
        tabLayout.addTab(tabLayout.newTab().setText("Dropit
Clean"));

        tabLayout.setTabGravity((TabLayout.GRAVITY_FILL));

        final ViewPager viewPager = (ViewPager)

```

```

findViewById(R.id.pager);
    final PagerAdapter adapter = new
PagerAdapter(getSupportFragmentManager(), tabLayout.getTabCount());
    viewPager.setAdapter(adapter);
    viewPager.setOnPageChangeListener(new
TabLayout.TabLayoutOnPageChangeListener(tabLayout));

    tabLayout.setOnTabSelectedListener(new
TabLayout.OnTabSelectedListener() {
        @Override
        public void onTabSelected(TabLayout.Tab tab) {
            viewPager.setCurrentItem(tab.getPosition());
        }

        @Override
        public void onTabUnselected(TabLayout.Tab tab) {

        }

        @Override
        public void onTabReselected(TabLayout.Tab tab) {

        }
    });
}

@Override
public void onFragmentInteraction(Uri uri) {

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home :
            Intent intent = new Intent(this, Home.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.putExtra("EXIT", true);
            startActivity(intent);
            overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
            finish();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

@Override
public void onBackPressed() {
    Intent intent = new Intent(this, Home.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

```

```

        intent.putExtra("EXIT", true);
        startActivity(intent);
        overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
        finish();
    }
}

```

## LOGIN MEMBER

```

package com.amar.shoes.shoescare;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.Typeface;
import android.inputmethodservice.Keyboard;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.NotificationCompat;
import android.text.Editable;
import android.text.Html;
import android.text.TextWatcher;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class LoginnActivity extends AppCompatActivity {

    RequestQueue requestQueue;
    StringRequest stringRequest;
    ProgressDialog pdialog;
    final String Urllogin =
"http://u312737575.hostingerapp.com/android/login2.php";
    Boolean session = false;

```



```

public static final String KEY_USERNAME="username";
public static final String KEY_PASSWORD="password";
public static final String KEY_NAMALENGKAP="nama_lengkap";
public static final String KEY_TELEPON="telepon";
public static final String KEY_ALAMAT="alamat";
public static final String KEY_PROFIL="fotoprofil";
public static final String KEY_ID="Id_username";

String namalengkap, alamat, telepon, fotoprofil, Id_username,
username, password;
SharedPreferences sharedPreferences;
public static final String Shared_Name ="My_login";
public static final String LOGIN ="Islogin";

private TextView text;
private EditText txtUser, txtPass;
private Button blogin;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    txtUser = (EditText) findViewById(R.id.txtuser);
    txtPass = (EditText) findViewById(R.id.txtpass);
    blogin = (Button) findViewById(R.id.btn_login);

    text = (TextView) findViewById(R.id.nameapp);
    Typeface customfont =
Typeface.createFromAsset(getAssets(), "fonts/angrybirds.ttf");
    text.setTypeface(customfont);
    text.setTextSize(20);

    txtUser.addTextChangedListener(LoginText);
    txtPass.addTextChangedListener(LoginText);

    getSupportActionBar().setDisplayShowTitleEnabled(false);
    getSupportActionBar().hide();
    txtPass.setOnKeyListener(new View.OnKeyListener() {
        @Override
        public boolean onKey(View v, int keyCode, KeyEvent
eventKey) {
            int ac = eventKey.getAction();
            if (ac == KeyEvent.ACTION_UP)
            {
                processbuttontext();
            }
            return false;
        }
    });

    txtUser.setOnKeyListener(new View.OnKeyListener() {
        @Override
        public boolean onKey(View view, int i, KeyEvent event)
{
            int action = event.getAction();
            if (action == KeyEvent.ACTION_UP)
            {

```

```

        processbuttonontext();
    }
    return false;
}
});

    sharedPreferences = getSharedPreferences(Shared_Name,
Context.MODE_PRIVATE);
    session = sharedPreferences.getBoolean(LOGIN, false);

    username = sharedPreferences.getString(KEY_USERNAME,
null);
    password = sharedPreferences.getString(KEY_PASSWORD,
null);
    Id_username= sharedPreferences.getString(KEY_ID, null);

    if (session) {
        Intent intent = new Intent(LoginnActivity.this,
PenggunaActivity.class);

        intent.putExtra(KEY_USERNAME, username);
        intent.putExtra(KEY_PASSWORD, password);
        intent.putExtra(KEY_ID, Id_username);

        startActivity(intent);
    }

    requestQueue = Volley.newRequestQueue(this);

    final Response.Listener listener = new
Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            JSONObject jsonObject;
            try {
                jsonObject = new JSONObject(response);
                boolean success =
jsonObject.getBoolean("success");

                if (success) {

                    String username =
jsonObject.getString(KEY_USERNAME);
                    String password =
jsonObject.getString(KEY_PASSWORD);
                    String Id_username =
jsonObject.getString(KEY_ID);

                    SharedPreferences.Editor editor =
sharedPreferences.edit();
                    editor.putBoolean(LOGIN, true);

                    editor.putString(KEY_USERNAME, username);
                    editor.putString(KEY_PASSWORD, password);

```

```

        editor.putString(KEY_ID, Id_username);

        editor.commit();

        Intent intent = new
Intent(LoginnActivity.this, PenggunaActivity.class);

        intent.putExtra(KEY_USERNAME, username);
        intent.putExtra(KEY_PASSWORD, password);
        intent.putExtra(KEY_ID, Id_username);

        startActivity(intent);

        notifTempllate(tittle,message);
    }else {
        AlertDialog.Builder builder = new
AlertDialog.Builder(LoginnActivity.this);
        builder.setMessage(Html.fromHtml("<font
color='#000'>Login Failed</font>"));
        builder.setNegativeButton("Retry",
null)

                .create()
                .show();
        pdialog.dismiss();

    }
    }catch (JSONException e) {
        e.printStackTrace();
        pdialog.dismiss();
    }
}
};

blogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        pdialog = new ProgressDialog(LoginnActivity.this);
        pdialog.setMessage("Sedang Memuat");
        pdialog.show();

        stringRequest = new
StringRequest(Request.Method.POST, Urllogin, listener, null) {
            @Override
            protected Map<String, String> getParams()
throws AuthFailureError {
                HashMap<String, String> hashmap = new
HashMap<String, String>();
                hashmap.put("username",
txtUser.getText().toString());
                hashmap.put("password",
txtPass.getText().toString());

                return hashmap;
            }
        };
    }
});

```

```

        }
    };

    requestQueue.add(stringRequest);

    }
});

}

private void notifTempllate(String tittle, String message) {
    final Intent intent = new Intent(LoginnActivity.this,
LoginnActivity.class);
    PendingIntent pendingIntent =
PendingIntent.getActivity(LoginnActivity.this, 15, intent,
PendingIntent.FLAG_UPDATE_CURRENT);

    NotificationCompat.Builder mbuilder =
(NotificationCompat.Builder) new
NotificationCompat.Builder(LoginnActivity.this)
        .setSmallIcon(R.mipmap.sulitttcare)
        .setContentTitle(tittle)
        .setContentText(message)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .setDefaults(Notification.DEFAULT_LIGHTS |
Notification.DEFAULT_SOUND | Notification.FLAG_AUTO_CANCEL);
    NotificationManager notificationmanager =
(NotificationManager)
LoginnActivity.this.getSystemService(Context.NOTIFICATION_SERVICE)
;
    notificationmanager.notify(1, mbuilder.build());
}

final String message = "Selamat Datang Member Shoes and
Care.";
final String tittle = "Member Shoes and Care";
private TextWatcher LoginText = new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start,
int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int
before, int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {
        processbuttontext();
    }
};

private void processbuttontext() {
    String inputre = txtUser.getText().toString().trim();
    String input = txtPass.getText().toString().trim();
}

```

```

        if (inputre.length() > 0 && input.length() > 0 )
        {
            blogin.setEnabled(true);
            blogin.setTextColor(Color.WHITE);

blogin.setBackgroundResource(R.drawable.btn_login_daftar);

        }else {
            blogin.setEnabled(false);
            blogin.setTextColor(Color.parseColor("#a1a1a1"));

blogin.setBackgroundResource(R.drawable.btn_login_daftars);
        }
    }

    public void pindahregister(View view) {
        Intent intent = new Intent(LoginnActivity.this,
Kedua.class);
        startActivity(intent);
    }

    public void backhome(View view) {

        Intent intent = new Intent(this, Home.class);
        startActivity(intent);
        overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
        finish();
    }
}

```

## REGISTER MEMBER

```

package com.amar.shoes.shoescare;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.NotificationCompat;
import android.text.Html;
import android.text.TextUtils;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

```

```

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.androidadvance.topsnackbar.TSnackbar;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Kedua extends AppCompatActivity {

    private EditText password, name, email;
    ProgressDialog progressDialog;
    private Button btn_daftar;
    private TextView text;

    private static String Url_daftar =
"http://u312737575.hostingerapp.com/android/register2.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kedua);

        name = (EditText) findViewById(R.id.txtdaftar);
        email = (EditText) findViewById(R.id.txtdalamat);
        btn_daftar = (Button) findViewById(R.id.btn_daftar);
        password = (EditText) findViewById(R.id.txtpassdaftar);

        getSupportActionBar().setDisplayShowTitleEnabled(false);
        getSupportActionBar().hide();

        text = (TextView) findViewById(R.id.nameapp);
        Typeface customfont =
Typeface.createFromAsset(getAssets(), "fonts/angrybirds.ttf");
        text.setTypeface(customfont);
        text.setTextSize(20);

        btn_daftar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String user = name.getText().toString().trim();
                String pass =
password.getText().toString().trim();
                String ema = email.getText().toString().trim();

```

```

        if (TextUtils.isEmpty(user)) {
            TSnackbar snack =
TSnackbar.make(findViewById(android.R.id.content), "Username Harus
Diisi", TSnackbar.LENGTH_LONG);
            View snackbarview = snack.getView();

snackbarview.setBackgroundColor(getResources().getColor(R.color.white));

            TextView tc =
(TextView) snackbarview.findViewById(com.androidadvance.topsnackbar.
R.id.snackbar_text);
            tc.setTextColor(Color.BLACK);
            snack.show();
            name.requestFocus();
            return;
        }
        if (TextUtils.isEmpty(ema)) {
            TSnackbar snack =
TSnackbar.make(findViewById(android.R.id.content), "Email Harus
Diisi", TSnackbar.LENGTH_LONG);
            View snackbarview = snack.getView();

snackbarview.setBackgroundColor(getResources().getColor(R.color.white));

            TextView tc =
(TextView) snackbarview.findViewById(com.androidadvance.topsnackbar.
R.id.snackbar_text);
            tc.setTextColor(Color.BLACK);
            snack.show();
            email.requestFocus();
            return;
        }
        if (TextUtils.isEmpty(pass)) {
            TSnackbar snack =
TSnackbar.make(findViewById(android.R.id.content), "Password Harus
Diisi", TSnackbar.LENGTH_LONG);
            View snackbarview = snack.getView();

snackbarview.setBackgroundColor(getResources().getColor(R.color.white));

            TextView tc =
(TextView) snackbarview.findViewById(com.androidadvance.topsnackbar.
R.id.snackbar_text);
            tc.setTextColor(Color.BLACK);
            snack.show();
            password.requestFocus();
            return;
        }
        else {
            regis(user, pass, ema);
        }
    }
});
}

private void regis(final String user, final String pass, final
String ema) {

```

```

        final ProgressDialog dialog = new ProgressDialog(this);
        dialog.setMessage("Sedang Memuat");
        dialog.show();

        StringRequest stringRequest = new
StringRequest(Request.Method.POST, Url_daftar,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new
JSONObject(response);

                    String success =
jsonObject.getString("success");

                    if (success.equals("1")){

                        Toast.makeText(getApplicationContext(), "Register success",
                        Toast.LENGTH_SHORT).show();

                        Intent i = new Intent(Kedua.this,
                        LoginnActivity.class);

                        startActivity(i);

                        notifTempllate(tittle,message);

                    }

                } catch (JSONException e) {
                    e.printStackTrace();

                    Toast.makeText(getApplicationContext(), "Register Error" +
                    e.toString(), Toast.LENGTH_SHORT).show();

                    dialog.dismiss();

                }

            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error)

        }

        {

            Toast.makeText(getApplicationContext(),
            "Register Error" + error.toString(), Toast.LENGTH_SHORT).show();

            dialog.dismiss();

        }

    }){

        @Override
        protected Map<String, String> getParams() throws
AuthFailureError {

            Map<String, String> Params = new HashMap<>();
            Params.put("username", user);
            Params.put("password", pass);
            Params.put("email", ema);
            return Params;
        }
    }

```



```

    }
};
RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.add(stringRequest);
}

private void notifTempllate(String tittle, String message) {
    final Intent intent = new Intent(Kedua.this,
    LoginnActivity.class);
    PendingIntent pendingIntent =
    PendingIntent.getActivity(Kedua.this, 15, intent,
    PendingIntent.FLAG_UPDATE_CURRENT);

    NotificationCompat.Builder mbuilder =
    (NotificationCompat.Builder) new
    NotificationCompat.Builder(Kedua.this)
        .setSmallIcon(R.mipmap.sulitttcare)
        .setContentTitle(tittle)
        .setContentText(message)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
        .setDefaults(Notification.DEFAULT_LIGHTS |
    Notification.DEFAULT_SOUND | Notification.FLAG_AUTO_CANCEL);
    NotificationManager notificationmanager =
    (NotificationManager)
    Kedua.this.getSystemService(Context.NOTIFICATION_SERVICE);
    notificationmanager.notify(1, mbuilder.build());
}

final String message = "Selamat Anda telah terdaftar Member
Shoes and Care.";
final String tittle = "Silahkan Login";
@Override
public void onBackPressed() {
    Intent intent = new Intent(this, LoginnActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.putExtra("EXIT", true);
    startActivity(intent);
    overridePendingTransition(R.anim.fade_in,
    R.anim.fade_out);
    finish();
}

public void backhome(View view) {
    Intent intent = new Intent(this, LoginnActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.putExtra("EXIT", true);
    startActivity(intent);
    overridePendingTransition(R.anim.fade_in,
    R.anim.fade_out);
    finish();
}
}
}

```