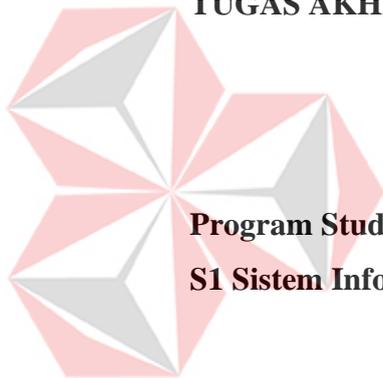




**RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN  
PADA BENGKEL MODIFIE MENGGUNAKAN METODE *MIN MAX***

**TUGAS AKHIR**



**Program Studi  
S1 Sistem Informasi**

UNIVERSITAS  
**Dinamika**

**Oleh :**

**YUSRIL RAHARDIAN**

**18.41010.0133**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

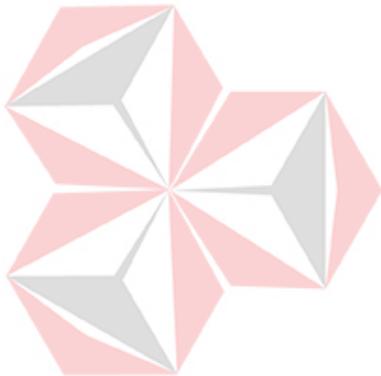
**UNIVERSITAS DINAMIKA**

**2022**

**RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN  
PADA BENGKEL MODIFIE MENGGUNAKAN METODE *MIN MAX***

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana**



UNIVERSITAS  
**Dinamika**

Oleh :

**Nama** : Yusril Rahardian  
**NIM** : 18.41010.0133  
**Program Studi** : S1 Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA**

**2022**

## TUGAS AKHIR

### RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN PADA BENGKEL MODIFIE MENGGUNAKAN METODE *MIN MAX*

Dipersiapkan dan disusun oleh

Yusril Rahardian

NIM: 18.41010.0133

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada: Kamis, 11 Agustus 2022

#### Susunan Dewan Pembahas

##### Pembimbing:

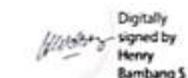
- I. Sulistiowati, S.Si., M.M  
NIDN. 0719016801
- II. Pantjawati Sudarmaningtyas, S.Kom., M.Eng  
NIDN. 0712066801

##### Pembahas:

- I. Ir. Henry Bambang Setyawan, M.M



Universitas Dinamika  
2022/08/12 11:40:58  
+07'00'



Digitally signed by  
Henry Bambang S

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana



Digitally signed by  
Universitas Dinamika  
Date: 2022.08.12  
17:48:15 +07'00'

Tri Sagranl, S.Kom., M.MT.

NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Yusril Rahardian  
NIM : 18410100133  
Program Studi : SI Sistem Informasi  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **RANCANG BANGUN APLIKASI PENGENDALIAN  
PERSEDIAAN PADA BENGKEL MODIFIE MENGGUNAKAN  
METODE MIN MAX**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 01 Juli 2022

Yang menyatakan



**Yusril Rahardian**

**NIM: 18410100133**



*Kehidupan tidaklah selalu indah, namun berjuanglah untuk hidup seindah  
mungkin sampai akhir, dengan begitu kau akan mendapatkan akhir yang indah :)*

- Sakata Gintoki (Gintama)

UNIVERSITAS  
Dinamika



*Daku persembahkan kepada*

*Keluarga,*

*Bapak Ibu Dosen,*

*Teman dan sahabat yang*

*Selalu memberikan dukungan dan semangat disetiap langkah*

UNIVERSITAS  
Dinamika

## ABSTRAK

Bengkel Modifie merupakan bengkel mobil yang berfokus pada perbaikan body mobil, pemeliharaan mobil, dan juga modifikasi mobil. Pada bengkel Modifie semua transaksi yang dilakukan masih dicatat melalui buku dan berkas-berkas secara manual. Terdapat permasalahan yaitu permintaan pelanggan yang tidak terpenuhi diakibatkan tidak ada *spare part* yang tersedia (kehabisan *spare part*) yang mengharuskan pelanggan untuk menunggu *spare part* untuk dipesan terlebih dahulu, dan juga pengecekan persediaan *inventory* harus dihitung secara langsung yang mengakibatkan data persediaan harus dilakukan pengecekan ulang satu per satu dan membutuhkan banyak waktu dan tenaga. Berdasarkan permasalahan yang ada bengkel Modifie, dibutuhkan aplikasi yang dapat membantu pencatatan transaksi *spare part*, dan juga pengendalian persediaan *spare part* agar dapat mencegah kehabisan stok *spare part*. Hal tersebut dapat diatasi dengan rancang bangun sebuah aplikasi yang dapat melakukan pengendalian persediaan dengan metode *Min Max* untuk menentukan titik *minimum stock* dan *maximum stock*. Berdasarkan dari hasil uji coba yang dilakukan didapatkan hasil bahwa aplikasi yang dibangun dapat mempersingkat *Service Advisor* dalam memberikan *list spare part* yang dibutuhkan Administrasi, melakukan rekap data dan laporan (pdf) pada transaksi *spare part* yaitu *spare part* keluar, pemesanan, penerimaan, dan retur *spare part*, aplikasi juga dapat memberikan notifikasi, daftar, dan jumlah *spare part* yang mendekati atau kurang dari nilai *minimum stock* sehingga dapat dilakukan pemesanan *spare part* sebelum stok habis, selain itu aplikasi dapat menentukan jumlah *spare part* yang harus dipesan sesuai dengan nilai *Q*. Aplikasi juga dapat memberikan peringatan stok *spare part* mendekati atau kurang dari nilai *minimum stock* ketika pengguna akan melakukan konfirmasi *spare part* keluar serta memberikan informasi apakah sudah dilakukan pemesanan atau tidak. Aplikasi juga dilengkapi dengan fitur jika stok *spare part* sama atau kurang dari jumlah *safety stock user* tidak bisa menginputkan *spare part* keluar (*unavailable*).

**Kata Kunci:** *Min Max*, Pengendalian Persediaan, Bengkel, *Spare Part*.

## KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Aplikasi Pengendalian Persediaan Pada Bengkel Modifie Menggunakan Metode Min Max”.

Penyelesaian Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak yang telah memberikan banyak masukan, nasihat, saran, kritik, dan dukungan moral maupun materil kepada penulis. Oleh karena itu penulis menyampaikan rasa terima kasih kepada:

1. Ibu, Bapak dan keluarga tercinta keluarga yang selalu mendoakan, mendukung, dan memberikan semangat di setiap langkah dan aktivitas penulis.
2. Bapak Prof. Dr. Budi Jatmiko, M.Pd. selaku Rektor Universitas Dinamika
3. Ibu Sulistiowati, S.Si., M.M, dan Ibu Pantjawati Surdamaningtyas, S.Kom., M.Eng selaku Dosen Pembimbing yang telah membimbing, mendukung dan memberi masukan dalam menyelesaikan laporan ini.
4. Bapak Ir. Henry Bambang Setyawan, M.M selaku Dosen Pembahas yang telah bersedia menjadi dosen pembahas dalam menyempurnakan laporan Tugas Akhir ini..
5. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng. selaku Ketua Program Studi S1 Sistem Informasi Universitas Dinamika
6. *Owner* Bengkel Modifie yang telah memberikan penulis kesempatan untuk dapat melakukan penelitian ini.
7. Teman-teman tercinta yang memberikan bantuan dan dukungan dalam menyelesaikan Tugas Akhir ini.
8. Pihak-pihak lain yang tidak dapat disebutkan satu-persatu yang telah memberikan bantuan dan dukungan kepada penulis.

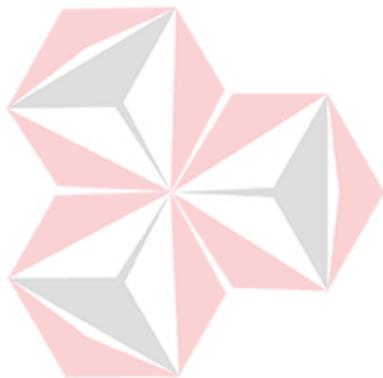
Semoga Allah SWT memberikan balasan yang setimpal kepada semua pihak yang telah membantu dan memberikan bimbingan serta nasehat dalam proses menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir yang dikerjakan ini masih banyak terdapat kekurangan sehingga kritik yang bersifat membangun dan saran dari semua

pihak sangatlah diharapkan agar aplikasi ini dapat diperbaiki menjadi lebih baik lagi. Semoga Tugas Akhir ini dapat diterima dan bermanfaat bagi penulis dan semua pihak.

Surabaya, 10 Agustus 2022

Penulis

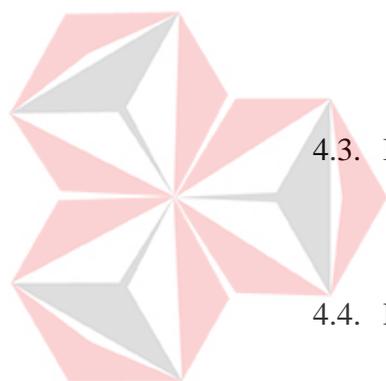


UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan .....	4
1.5. Manfaat .....	4
BAB II LANDASAN TEORI .....	5
2.1. Penelitian Terdahulu .....	5
2.2. Persediaan .....	6
2.3. Pengendalian Persediaan .....	6
2.4. Metode <i>Min Max</i> .....	6
2.5. <i>Safety Stock</i> .....	7
2.6. <i>System Development Life Cycle (SDLC)</i> .....	8
2.7. <i>Black Box Testing</i> .....	9
2.8. <i>User Acceptance Testing (UAT)</i> .....	10
BAB III METODOLOGI PENELITIAN .....	11
3.1. <i>Communication</i> .....	12
3.1.1. <i>Project Initiation</i> .....	12
3.1.2. <i>Requirement Gathering</i> .....	13
3.2. <i>Planning</i> .....	14
3.3. <i>Modeling</i> .....	15
3.4. <i>Construction</i> .....	15
3.4.1. <i>Coding</i> .....	15

3.4.2. <i>Testing</i> .....	15
BAB IV HASIL PEMBAHASAN .....	21
4.1. Hasil <i>Requirement Gathering</i> .....	21
4.1.1. Hasil Analisis Proses Bisnis .....	21
4.1.2. Hasil Kebutuhan Pengguna .....	24
4.1.3. Hasil Kebutuhan Fungsional .....	25
4.1.4. Hasil Kebutuhan Nonfungsional .....	26
4.1.5. Hasil Kebutuhan Sistem .....	26
4.1.6. Hasil Diagram IPO .....	27
4.2. Hasil Desain.....	33
4.2.1. Hasil <i>Sysflow</i> .....	33
4.2.2. Hasil <i>Data Flow Diagram (DFD)</i> .....	34
4.2.3. Hasil <i>Conceptual Data Model (CDM)</i> .....	37
4.2.4. Hasil <i>Physical Data Model (PDM)</i> .....	38
4.2.5. Hasil <i>Desain Interface</i> .....	38
4.3. Implementasi Sistem .....	39
4.3.1. Implementasi <i>Service Advisor (SA)</i> .....	39
4.3.2. Implementasi Administrasi .....	40
4.4. Hasil <i>Testing</i> .....	44
4.4.1. <i>Black Box Testing</i> .....	45
4.4.2. <i>User Acceptance Testing (UAT)</i> .....	45
BAB V PENUTUP.....	47
5.1. Kesimpulan.....	47
5.2. Saran .....	47
DAFTAR PUSTAKA .....	49
LAMPIRAN.....	51



## DAFTAR TABEL

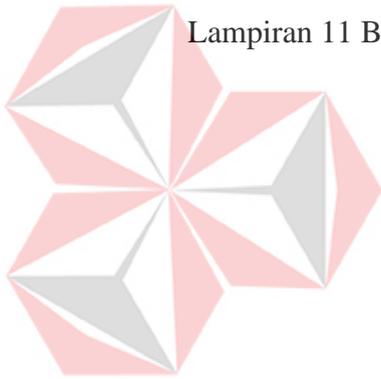
	Halaman
Tabel 1.1 Data Mobil Modifie .....	2
Tabel 2.1 Penelitian Terdahulu .....	5
Tabel 3.1 Skenario <i>Black Box Testing Login</i> .....	16
Tabel 3.2 Skenario <i>Black Box Testing (SA)</i> .....	16
Tabel 3.3 Skenario <i>Black Box Testing (Administrasi)</i> .....	16
Tabel 3.4 Skenario UAT (SA) .....	17
Tabel 3.5 Skenario UAT (Administrasi).....	18
Tabel 4.1 Identifikasi Masalah .....	22
Tabel 4.2 Kebutuhan Pengguna <i>Service Advisor (SA)</i> .....	24
Tabel 4.3 Kebutuhan Pengguna Bagian Administrasi .....	24
Tabel 4.4 Kebutuhan Fungsional .....	25
Tabel 4.5 Kebutuhan Sistem .....	26
Tabel 4.6 <i>Black Box Testing</i> .....	45
Tabel 4.7 Contoh Perhitungan Manual Implementasi.....	45
Tabel 4.8 <i>User Acceptance Testing (Service Advisor)</i> .....	45
Tabel 4.9 <i>User Acceptance Testing (Administrasi)</i> .....	46

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model <i>Waterfall</i> .....	8
Gambar 3.1 Metodologi Penelitian .....	11
Gambar 4.1 Diagram IPO (1).....	27
Gambar 4.2 Diagram IPO (2).....	28
Gambar 4.3 <i>Sysflow</i> (Utama) .....	34
Gambar 4.4 <i>Context</i> Diagram .....	35
Gambar 4.5 DFD Level 0.....	36
Gambar 4.6 CDM.....	38
Gambar 4.7 PDM .....	38
Gambar 4.8 Desain <i>Dashboard</i> .....	39
Gambar 4.9 Desain Hasil Perhitungan <i>Min Max</i> .....	39
Gambar 4.10 Implementasi <i>Spare Part</i> Keluar 3 (Administrasi) .....	41
Gambar 4.11 Implementasi Hasil Perhitungan <i>Min Max</i> 1.....	42
Gambar 4.12 Implementasi Hasil Perhitungan <i>Min Max</i> 2.....	43
Gambar 4.13 Implementasi Hasil Perhitungan <i>Min Max</i> 3.....	43
Gambar 4.14 Implementasi Hasil Perhitungan <i>Min Max</i> 4.....	43
Gambar 4.15 Implementasi Hasil Perhitungan <i>Min Max</i> 5.....	44
Gambar 4.16 Pemesanan Melalui Pengendalian Persediaan.....	44

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Jadwal Kerja .....	51
Lampiran 2 Analisis Proses Bisnis.....	52
Lampiran 3 <i>Sysflow</i> .....	53
Lampiran 4 Diagram Berjenjang.....	54
Lampiran 5 <i>Data Flow Diagram</i> (DFD).....	55
Lampiran 6 Desain <i>Interface</i> .....	62
Lampiran 7 Implementasi Sistem.....	80
Lampiran 8 <i>Black Box Testing</i> .....	120
Lampiran 9 <i>User Acceptance Testing</i> .....	127
Lampiran 10 Hasil Plagiasi .....	138
Lampiran 11 Biodata Penulis .....	144



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pada saat ini dimana teknologi berkembang pesat, kebutuhan akan teknologi pun semakin meningkat, seperti bagaimana teknologi dibutuhkan dalam kehidupan sehari-hari. Salah satu contohnya adalah dalam pengelolaan data dan pengendalian persediaan (*inventory*), seperti mengelola data *inventory*, yang termasuk keluar masuknya barang dari pelanggan ataupun *supplier*, berapa jumlah stok yang harus di *restock* atau berapa jumlah stok yang harus dipesan yang diketahui dari jumlah dan juga laporan yang dapat memberikan informasi terperinci mengenai transaksi pada *spare part*.

Bengkel Modifie merupakan bengkel mobil yang didirikan dari tahun 2004 dan berlokasi di Jl. Raya Tenggilis Mejoyo 14 / F12 Surabaya, dimana pada bengkel ini bisa dilakukan perbaikan body mobil, pemeliharaan mobil, dan juga modifikasi mobil. Bengkel Modifie sendiri berfokus pada modifikasi mobil pada *speedometer*, *panel*, dan *headlamp* sesuai permintaan pelanggan. Bengkel Modifie sendiri memiliki karyawan yang berjumlah 20 karyawan, meliputi 3 Administrasi, 1 Kepala Mekanik, 6 Mekanik, 2 *Service Advisor*, 4 Teknisi Cat, 1 *Supervisor* bengkel, 1 *Security*, 1 *Cleaning Service*, 1 *Graphic Designer*.

Proses bisnis yang dilakukan pada bengkel Modifie masih dilakukan secara konvensional, seperti penerimaan keluhan mobil, pencatatan *spare part*, penggajian pegawai, status perbaikan mobil, dan antrian pelanggan. Pencatatan *spare part* yaitu berupa keluar masuknya *spare part*, keluarnya *spare part* sendiri terjadi disaat mekanik membutuhkan *spare part* untuk memperbaiki mobil, sedangkan masuknya *spare part* adalah saat dilakukan pemesanan dan penerimaan *spare part* dari *supplier*, untuk pemesanan sendiri masih dilakukan melalui telpon, *gmail* atau aplikasi *whatsapp*. Semua transaksi yang dilakukan pada bengkel Modifie masih dicatat melalui buku dan berkas-berkas secara manual.

Masalah yang terjadi pada bagian persediaan (*inventory*) yaitu pertama permintaan pelanggan yang tidak terpenuhi diakibatkan tidak ada *spare part* yang

tersedia (kehabisan *spare part*) yang mengharuskan pelanggan untuk menunggu *spare part* untuk dipesan terlebih dahulu, kedua pengecekan persediaan (*inventory*) harus dihitung secara langsung yang mengakibatkan data persediaan harus dilakukan pengecekan ulang satu per satu dan membutuhkan banyak waktu dan tenaga.

Data bulan Januari sampai Agustus 2021 menunjukkan rata-rata mobil masuk bengkel sekitar 284 mobil, dan rata-rata jumlah mobil yang keluar (mobil yang sudah diperbaiki) 255. Dari jumlah tersebut sekitar 10% pelanggan tidak mendapat *spare part* yang sesuai, karena stok tidak tersedia dan pelanggan menjadi pergi meninggalkan bengkel karena tidak ingin menunggu sampai *spare part* datang, untuk data lebih rinci bisa dilihat pada Tabel 1.1.

Tabel 1.1 Data Mobil Modifie

Periode 2021	Jml Mobil Masuk	Jml Mobil Keluar
Januari	300	270
Februari	278	245
Maret	301	287
April	299	280
Mei	261	223
Juni	302	281
Juli	233	202
Agustus	300	253
<b>Rata – Rata</b>	<b>284.25</b>	<b>255.125</b>

Saat ini proses manajemen persediaan (*inventory*) dilakukan dengan mencatat transaksi pengeluaran maupun pemasukan *spare part* ke dalam buku. Hal ini menimbulkan kesulitan karena persediaan *spare part* tidak dapat dipantau secara *real time*. Pengecekan persediaan (*inventory*) harus dihitung secara langsung, sehingga membutuhkan banyak waktu dan tenaga. Selain itu, hal ini juga mengakibatkan jumlah pemesanan *spare part* tidak jelas dan *restock spare part* menjadi kurang maksimal karena tidak adanya metode yang membantu dalam pengendalian persediaan (*inventory control*) yang mengakibatkan kesulitan menentukan kapan *spare part* harus dibeli dan dalam jumlah berapa yang optimal agar dapat memenuhi permintaan pelanggan.

Solusi yang ditawarkan adalah membuat aplikasi yang dapat mendukung dan membantu mengelola data persediaan (*inventory*), serta mempermudah proses

transaksi dan manajemen persediaan yang ada. Aplikasi yang akan diterapkan harus dirancang dengan baik agar bisa mengurangi terjadinya kesalahan pada transaksi. Aplikasi ini nantinya dilengkapi dengan pengendalian persediaan metode *Min Max* agar dapat mencegah terjadinya kekurangan atau kehabisan stok (*out of stock*) dan mencegah terjadinya kelebihan stok (*overstock*). Alasan digunakan metode ini karena cocok dengan persediaan pada bengkel Modifie karena bisa digunakan untuk menghindari *out of stock* agar dapat mengurangi jumlah mobil masuk yang tidak mendapat *spare part* yang sesuai. Aplikasi ini diharapkan dapat mengelola data persediaan, transaksi (permintaan, pemesanan, penerimaan, *spare part* keluar, dan retur barang), juga untuk menghasilkan informasi yang dibutuhkan seperti laporan stok dan keluar masuknya *spare part*.

## 1.2. Rumusan Masalah

Berdasarkan penjelasan latar belakang diatas, dapat dirumuskan masalah penelitian ini adalah bagaimana merancang dan membangun aplikasi pengendalian persediaan pada bengkel Modifie menggunakan metode *Min Max*.

## 1.3. Batasan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat disampaikan bahwa batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi yang dibuat hanya berfokuskan pada bagian persediaan (*inventory*).
2. Metode yang digunakan adalah *Min Max* yang dapat untuk menghitung *safety stock*, batasan stok persediaan minimal, maksimal, dan jumlah yang harus dipesan ulang agar mencapai batas maksimum.
3. Untuk *lead time* dalam satuan hari.
4. Aplikasi membahas mengenai permintaan, pemesanan, penerimaan, pengeluaran, pengendalian persediaan, dan retur *spare part*.
5. Aplikasi tidak membahas peramalan maupun pendukung keputusan dalam pengadaan *spare part*.
6. Laporan yang dihasilkan adalah laporan pemesanan *spare part*, penerimaan *spare part*, *spare part* keluar, dan retur *spare part*.
7. *User* atau pengguna sistem ini adalah pihak internal dari bengkel Modifie.

8. Aplikasi berbasis *website*.

#### 1.4. Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah, maka dapat disampaikan bahwa tujuan dari penelitian ini adalah menghasilkan aplikasi pengendalian persediaan (*inventory*) dengan metode *Min Max* sehingga mencegah terjadinya kekurangan stok *spare part* dengan menentukan jadwal pembelian stok *spare part*.

#### 1.5. Manfaat

Berikut merupakan manfaat penerapan aplikasi pengelolaan persediaan *spare part* menggunakan metode *Min Max* antara lain:

1. Membantu dalam pengendalian stok *spare part*.
2. Membantu dalam proses rekap data *spare part*.
3. Membantu dalam pembuatan laporan *spare part* keluar, pemesanan, penerimaan, dan retur.



UNIVERSITAS  
Dinamika

## BAB II

### LANDASAN TEORI

#### 2.1. Penelitian Terdahulu

Berikut merupakan penelitian terdahulu berupa jurnal terkait dengan penelitian yang dilakukan penulis dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

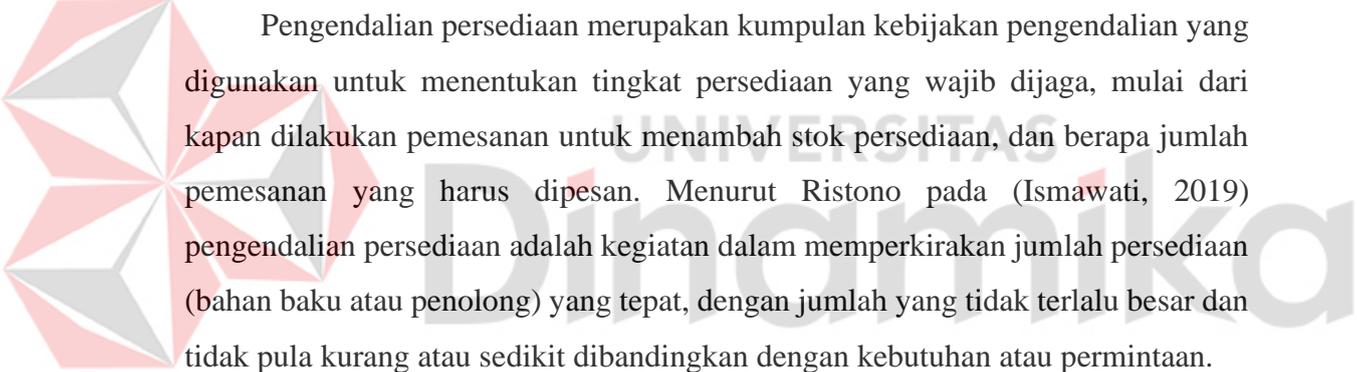
Nama Peneliti	Judul Penelitian	Hasil Penelitian
<b>(Leidiyana &amp; Anugrah, 2020)</b>	Aplikasi Pengendalian Persediaan Barang Berbasis Android Dengan Metode <i>Economic Order Quantity</i> (EOQ) Pada Bengkel Dunia Motor	Pada jurnal ini dikatakan bahwa implementasi aplikasi dapat membantu untuk mengetahui atau memperoleh jumlah pemesanan yang optimal berdasarkan biaya persediaan ( <i>inventory</i> ).
<b>Perbedaan : Pada penelitian ini menggunakan metode EOQ sedangkan pada penelitian yang akan dilakukan digunakan metode <i>Min Max</i></b>		
<b>(Sofi'i, Setianti, &amp; Purbasari, 2020)</b>	Sistem Informasi Akuntansi Pengendalian Persediaan Pada Sejati Bengkel Purwokerto	Pada jurnal ini dikatakan bahwa Modul aplikasi yang dikembangkan, semuanya dapat berfungsi dengan baik dan mampu menyajikan informasi sesuai yang diharapkan, Semua proses persediaan masuk dan keluar dilakukan terkomputerisasi sehingga sistem akan betul-betul menjadi <i>tool</i> pengendali persediaan, Dengan terkendalinya persediaan, akan dapat meningkatkan profitabilitas perusahaan.
<b>Perbedaan : Pada penelitian ini tidak digunakan metode pengendalian persediaan, hanya Sistem Informasi persediaan yang terintegrasi oleh sistem, namun pada penelitian yang akan dilakukan akan digunakan metode <i>Min Max</i>.</b>		
<b>(Wahyudin &amp; Kristiyanto, 2019)</b>	Rancang Bangun Sistem Informasi Persediaan <i>Spare Part</i> Mobil Berbasis <i>Web</i> Menggunakan Metode <i>Abc</i>	Pada jurnal ini dijelaskan aplikasi yang dibuat dapat membantu pihak perusahaan dalam menjalankan proses bisnis dengan <i>Standard Operating Prosedures</i> (SOP) yang baik dan benar, dan dengan adanya pengelompokan <i>spare part</i> pada Bengkel Tubagus Ban dapat membantu perusahaan untuk lebih memperhatikan mana yang menjadi prioritas utama pada jenis <i>spare part</i> yang memiliki nilai investasi paling tinggi.
<b>Perbedaan : Pada penelitian ini digunakan metode ABC sedangkan pada penelitian yang akan dilakukan digunakan metode <i>Min Max</i></b>		

## 2.2. Persediaan

Persediaan atau *inventory* menurut Steveson dalam (Naim & Donoriyanto, 2020) merupakan simpanan barang atau stok yang disimpan perusahaan yang berhubungan dengan bisnis yang dilakukan. Sedangkan menurut Sritomo pada (Nisa, 2019) merupakan “timbunan” barang (bahan baku, produk setengah jadi, atau produk akhir, dan lain-lain) yang secara sengaja disimpan sebagai cadangan (*safety stock*) untuk menghadapi kelangkaan pada saat proses produksi berlangsung.

Jadi persediaan adalah sebuah barang yang disimpan yang nantinya akan digunakan oleh perusahaan, untuk dijual, atau dibuat untuk bahan baku produksi yang digunakan untuk memenuhi permintaan dari pelanggan.

## 2.3. Pengendalian Persediaan



Pengendalian persediaan merupakan kumpulan kebijakan pengendalian yang digunakan untuk menentukan tingkat persediaan yang wajib dijaga, mulai dari kapan dilakukan pemesanan untuk menambah stok persediaan, dan berapa jumlah pemesanan yang harus dipesan. Menurut Ristono pada (Ismawati, 2019) pengendalian persediaan adalah kegiatan dalam memperkirakan jumlah persediaan (bahan baku atau penolong) yang tepat, dengan jumlah yang tidak terlalu besar dan tidak pula kurang atau sedikit dibandingkan dengan kebutuhan atau permintaan.

## 2.4. Metode *Min Max*

Pada metode *Min Max*, kuantitas minimal dan maksimal harus ditentukan terlebih dahulu, tingkatan minimal pada metode ini adalah batas pengaman yang digunakan untuk mencegah terjadinya kekurangan stok, dari nilai tingkat minimal ini juga merupakan titik untuk melakukan *reorder* atau pemesanan kembali, dimana kuantitas bahan baku yang dipesan adalah sebesar kebutuhan agar membuat persediaan pada tingkat yang maksimum. Menurut (Widiyanto, 2021) "*Min*" merupakan nilai tingkat persediaan yang memicu pemesanan ulang dan "*Max*" merupakan nilai tingkat persediaan baru yang ditargetkan mengikuti pemesanan ulang tersebut.

Rumus *Min Max* menurut Indrajit dan Djokopranoto dalam (Prabawa,

Darmawiguna, & Wirawan, 2018) adalah sebagai berikut :

$$Q = Max - Min \dots\dots\dots(1)$$

$$Min = (K \times W) + SS \dots\dots\dots(2)$$

$$Max = 2(K \times W) \dots\dots\dots(3)$$

Keterangan :

- Q : Jumlah yang perlu dipesan untuk pengisian persediaan kembali.  
 Min : *Minimum Stock*, adalah jumlah pemakaian selama waktu pesanan pembelian yang dihitung dari perkalian antara waktu pesanan perbulan dan pemakaian rata rata perbulan ditambah dengan persediaan pengaman.  
 Max : *Maximum Stock*, adalah jumlah maksimum yang diperbolehkan disimpan dalam persediaan, yaitu jumlah pemakaian selama 2 x waktu pesanan, yang dihitung dari perkalian antara 2 x waktu pesanan dan pemakaian rata-rata ditambah dengan persediaan pengaman.  
 K : Permintaan barang rata-rata  
 W : Waktu pesanan dalam satuan waktu (*lead time*) hitungan hari  
 SS : Jumlah persediaan pengaman (*safety stock*)

## 2.5. Safety Stock

*Safety stock* adalah sejumlah unit persediaan yang ditambahkan dalam pembelian persediaan yang ekonomis yang digunakan untuk penjagaan atas permintaan pelanggan yang tidak umum atau *lead time* yang lama (Ryando & Susanti, 2019). Sedangkan menurut Hardjanto pada (Ryando & Susanti, 2019) *safety stock* adalah persediaan yang berfungsi untuk melindungi atau menjaga kemungkinan terjadinya kekurangan barang, misalnya karena penggunaan barang yang lebih besar dari perkiraan semula atau keterlambatan dalam penerimaan barang yang dipesan.

Rumus *safety stock* menurut (Siboro & Nasution, 2020) adalah sebagai berikut :

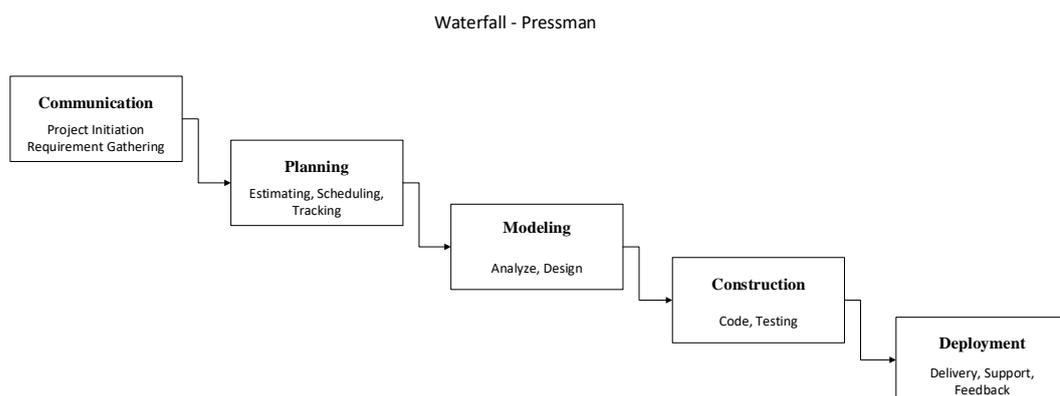
$$SS = (Permintaan maksimum - K) \times W \dots\dots\dots(4)$$

Keterangan:

- SS : Jumlah persediaan pengaman (*safety stock*)  
 K : Permintaan barang rata-rata  
 W : Waktu pesanan dalam satuan waktu / *lead time* (hari)

## 2.6. System Development Life Cycle (SDLC)

Siklus hidup pengembangan sistem (*System Development Life Cycle*) atau SDLC adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer, yang memiliki tahapan-tahapan *Planning*, Analisis, Desain, dan Implementasi (Permana & Sahara, 2018). Sedangkan menurut Azhar Susanto pada (Bolung & Tampangela, 2017) menyatakan bahwa SDLC (*System Development Life Cycle*) adalah salah satu metode pengembangan sistem informasi yang populer pada saat sistem informasi pertama kali dibuat. Sedangkan menurut Pressman pada (Amrin, Larasati, & Satriadi, 2020) model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Untuk gambar urutan tahapan *waterfall* menurut Pressman pada (Amrin, Larasati, & Satriadi, 2020) dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model *Waterfall*  
(Sumber gambar penulis, referensi dari Pressman pada (Amrin, Larasati, & Satriadi, 2020))

Beberapa tahapan metode yang dijelaskan Pressman:

a. *Communication (Project Initiation, Requirements Gathering)*

Tahap pertama dari metode *waterfall* adalah komunikasi (*Communication*). Tahap ini merupakan tahap pengumpulan informasi tentang masalah yang dihadapi dari pelanggan. Informasi didapat dari proses seperti wawancara, dokumen-dokumen, observasi, artikel, dan studi literatur. Hasil dari tahap ini adalah inisiasi proyek, analisis masalah dan informasi terkait tentang masalah yang dihadapi.

b. *Planning (Jadwal Kerja)*

Tahap berikutnya adalah tahap perencanaan (*Planning*). Tahap ini adalah tahap yang berisi tentang perkiraan aktifitas atau pun tugas yang akan dilakukan selanjutnya, segala resiko yang dapat terjadi, menjelaskan estimasi, penjadwalan kerja dan tracking progress dari pekerjaan ini.

c. *Modeling*

Tahap ini adalah tahap perancangan model (*Modeling*) dari sistem yang akan dikerjakan. Tahap ini berfokus pada struktur data, arsitektur *software* atau komposisi aplikasi, desain antarmuka.

d. *Construction (Code, Testing)*

Tahap ini adalah tahap pembangunan atau pembuatan aplikasi (*coding*) dari tahap sebelumnya. Setelah itu dilakukan pengecekan (*testing*) kepada hasil aplikasi yang telah dibuat agar dapat menemukan kesalahan, *bug*, dan kekurangan saat *testing* dilakukan yang nantinya akan diperbaiki.

e. *Deployment (Delivery, Support, Feedback)*

Tahap deployment merupakan tahap implementasi aplikasi yang telah dihasilkan ke pelanggan, yang nantinya juga dilakukan pemeliharaan pada aplikasi, evaluasi aplikasi, perbaikan, dan pengembangan aplikasi dari pengalaman pengguna saat menggunakan aplikasi.

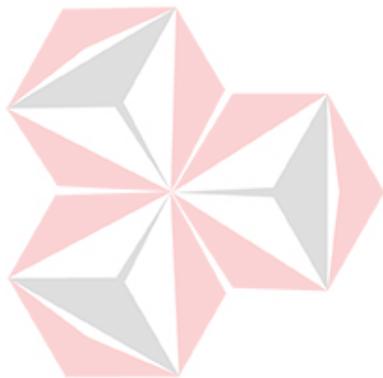
## 2.7. *Black Box Testing*

Menurut (Aulia & Indrayanti, 2021) *black box testing* adalah pengujian yang tidak memerlukan pengetahuan internal dari aplikasi, pengujian ini hanya memeriksa aspek fundamental dari sistem dan tidak memiliki atau hanya sedikit

relevansi dengan internal struktur logis dari sistem.

### **2.8. *User Acceptance Testing (UAT)***

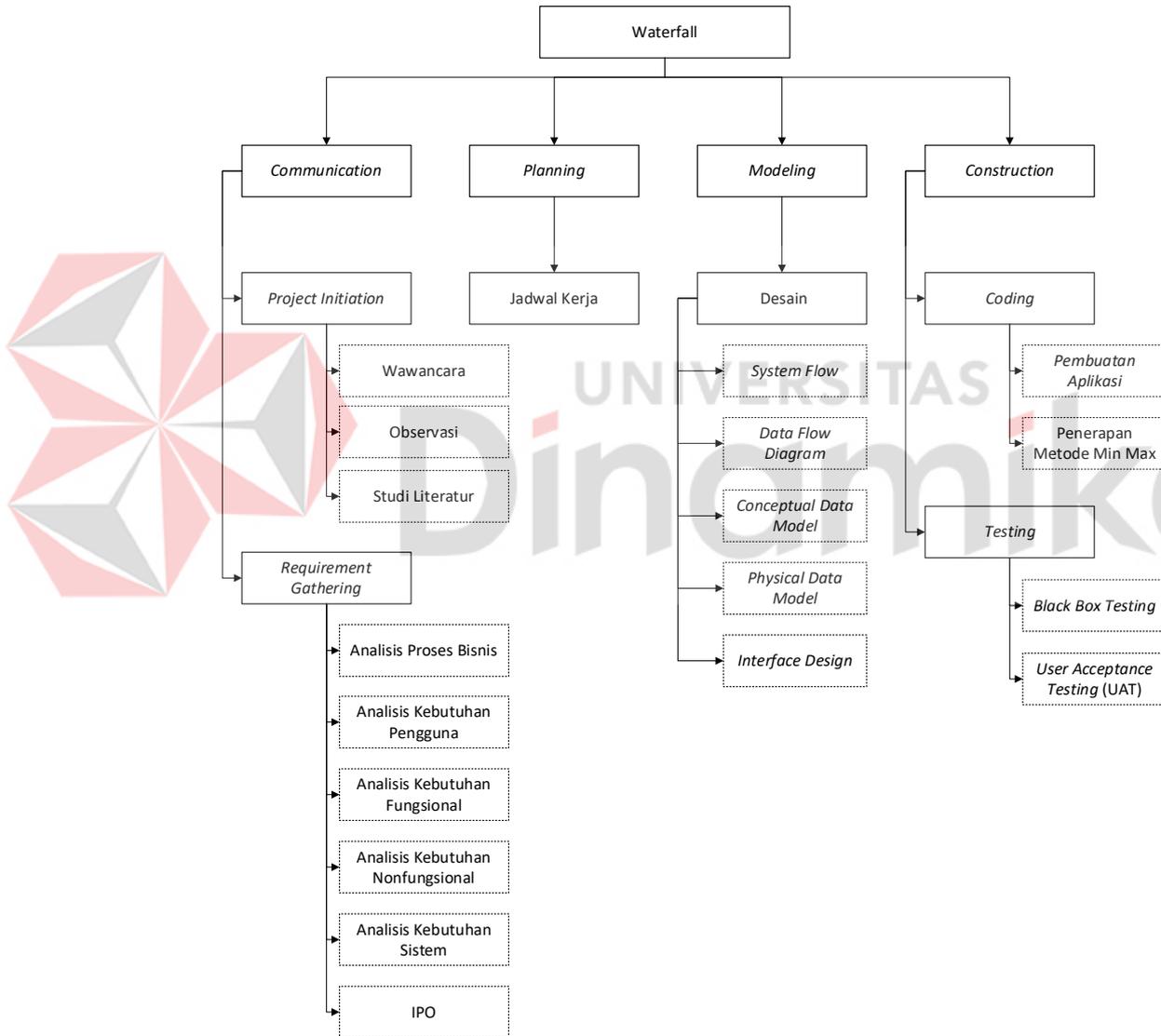
UAT merupakan sebuah uji coba terhadap sebuah *software* sebelum diluncurkan dengan memberikan sebuah kuisioner kepada *user* mengenai fungsi pada *software* apakah sudah bisa berjalan dengan baik atau tidak, singkatnya untuk validasi bahwa aplikasi atau *software* yang dibuat sesuai dengan kebutuhan *user*. Sedangkan menurut (Hady, Haryono, & Rahayu, 2019) UAT dilaksanakan pada akhir proses pengujian saat sistem siap digunakan. Tujuan utamanya adalah untuk mengembangkan perangkat lunak yang mampu memenuhi kebutuhan pengguna.



UNIVERSITAS  
**Dinamika**

### BAB III METODOLOGI PENELITIAN

Metode penelitian yang akan digunakan pada penelitian ini adalah *System Development Life Cycle (SDLC) Waterfall*. Metode ini digunakan sebagai landasan pada perancangan dan pembangunan aplikasi atau sistem, Tahapan yang akan diterapkan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian

### **3.1. Communication**

Tahap pertama merupakan *Communication* yaitu untuk berkomunikasi kepada *owner* Bengkel Modifie untuk mengumpulkan informasi tentang kendala dan alur pada pengelolaan persediaan bengkel dan data-data yang dibutuhkan. Terdapat dua kegiatan dalam tahap *Communication* yaitu *Project Initiation* dan *Requirement Gathering*.

#### **3.1.1. Project Initiation**

Sesuai pada Gambar 3.1 Metodologi Penelitian, bahwa di tahap *project initiation* ada tiga tahap, yaitu Wawancara, Observasi, dan Studi Literatur.

##### **A. Wawancara**

Dilakukan wawancara kepada pemilik bengkel mengenai masalah yang dialami pada bengkel Modifie yang dijalankan dengan mendatangi tempat langsung. Sebelum melakukan wawancara dipersiapkan daftar pertanyaan sesuai dengan teori yang digunakan, seperti alur proses bisnis, masalah yang ada, dan struktur karyawan bengkel, pertanyaan yang diajukan pada wawancara bersifat tertutup karena ditanyakan mengenai sistem dan alur proses bisnis pada bengkel, selain itu dilakukan wawancara kepada beberapa karyawan bengkel (bagian persediaan dan Administrasi) mengenai jasa yang dilakukan dan masalah yang ada.

##### **B. Observasi**

Observasi dilakukan untuk mengamati proses bisnis yang ada pada bengkel Modifie, untuk mendapatkan informasi dan data yang berhubungan dengan penyelesaian masalah yang tidak didapat melalui hasil wawancara.

##### **C. Studi Literatur**

Selain itu juga dilakukan studi literatur yaitu dengan mencari referensi melalui jurnal dan buku. Beberapa referensi tersebut berisikan mengenai:

1. Penelitian Terdahulu
2. Persediaan
3. Pengendalian Persediaan

4. *Min Max*
5. *Safety Stock*
6. *User Acceptance Testing (UAT)*
7. *Black Box Testing*
8. *System Development Life Cycle (SDLC)*

### **3.1.2. Requirement Gathering**

Tahap *requirement gathering* merupakan tahapan yang digunakan peneliti untuk menganalisa kebutuhan pembangunan sistem sesuai dengan hasil dari wawancara yang telah dilakukan pada tahap *project initiation*. Pada tahap ini peneliti melakukan analisis proses bisnis, analisis kebutuhan pengguna, analisis kebutuhan fungsional, analisis kebutuhan nonfungsional, analisis kebutuhan sistem, dan diagram IPO.

#### **A. Analisis Proses Bisnis**

Analisis proses bisnis didapatkan dari hasil wawancara dan observasi yang telah dilakukan, pada analisis proses bisnis nantinya akan dijelaskan mengenai kondisi proses bisnis yang terjadi saat ini pada bengkel Modifie seperti identifikasi masalah, identifikasi pengguna, identifikasi kebutuhan fungsional, dan identifikasi kebutuhan data.

##### **1. Identifikasi Masalah**

Pada tahapan ini bertujuan untuk mendapatkan permasalahan yang ada, dampak dari permasalahan, dan solusi yang ditawarkan untuk masalah tersebut.

##### **2. Identifikasi Pengguna**

Pada tahapan ini dilakukan proses identifikasi pengguna yang nantinya digunakan untuk proses analisis kebutuhan pengguna.

##### **3. Identifikasi Kebutuhan Fungsional**

Pada tahapan ini dilakukan proses identifikasi kebutuhan fungsional yang nantinya digunakan untuk proses analisis kebutuhan fungsional.

##### **4. Identifikasi Data**

Pada tahapan ini dilakukan proses identifikasi data yang nantinya digunakan untuk pembuatan diagram IPO.

## **B. Analisis Kebutuhan Pengguna**

Analisis kebutuhan pengguna ini digunakan untuk menggambarkan keterkaitan antara pengguna yang akan menggunakan aplikasi ini dengan aplikasi. Kebutuhan pengguna didapatkan dari hasil wawancara dan analisis proses bisnis.

## **C. Analisis Kebutuhan Fungsional**

Kebutuhan fungsional merupakan analisis terhadap fungsional sistem yang akan dibuat untuk tiap pengguna aplikasi nantinya (Administrasi dan *Service Advisor* (SA)) yang akan dijelaskan pada kebutuhan fungsional adalah aktor yang berhubungan pada fungsi-fungsi yang ada dan deskripsi fungsi tersebut.

## **D. Analisis Kebutuhan Nonfungsional**

Kebutuhan nonfungsional dimaksudkan untuk melakukan analisis kebutuhan nonfungsional yang dibutuhkan oleh sistem. Kebutuhan fungsional didapatkan dari hasil wawancara dan hasil analisis yang dilakukan sebelumnya.

## **E. Analisis Kebutuhan Sistem**

Pada tahap ini peneliti menjelaskan analisa kebutuhan setiap sistem yang akan dibangun. Kebutuhan sistem didapatkan dari hasil aktivitas yang dilakukan sebelumnya, yaitu wawancara dan analisis yang dilakukan.

## **F. Diagram IPO**

Diagram IPO didapatkan dari analisis kebutuhan yang telah dilakukan oleh peneliti, pada IPO akan dijelaskan mengenai *input*, *proses*, dan *output* yang ada pada sistem nantinya.

## **3.2. Planning**

*Planning* berisi kegiatan pembuatan jadwal pada proses pengerjaan sistem yang akan dilakukan penulis. Hasil dari tahap ini berupa jadwal kerja yang dapat dilihat pada Lampiran 1.

### 3.3. Modeling

Pada tahap *Modeling* merupakan tahapan yang menggambarkan bagaimana aplikasi yang akan dibangun nantinya. Pada *Modeling* akan dilakukan pembuatan *Sysflow*, *Data Flow Diagram* (DFD), *Conceptual Data Model* (CDM), *Physical Data Model* (PDM), dan *Desain Interface*. *Sysflow* merupakan alur dari proses bisnis aplikasi yang akan dibuat nantinya, *sysflow* dibuat menggunakan *tool microsoft visio*. *Data flow diagram* (DFD) digunakan penulis untuk menggambarkan alur data pada aplikasi, untuk pembuatan *data flow diagram* digunakan *tool power designer*. Perancangan *conceptual data model* (CDM) dilakukan oleh penulis setelah dirancangnya *data flow diagram*, perancangan *conceptual data model* (CDM) digunakan *tool power designer*. Perancangan *physical data model* (PDM) dilakukan oleh penulis setelah dirancangnya *conceptual data model* (CDM), perancangan *physical data model* (PDM) digunakan *tool power designer*. Setelah dirancang *sysflow*, *data flow diagram*, diagram berjenjang, *conceptual data model*, dan *physical data model* berlanjut pada tahap *desain interface*, pada tahap ini peneliti menggambarkan *desain interface* dari aplikasi yang akan dibuat menggunakan *tool adobe xd*.

### 3.4. Construction

Pada tahap ini peneliti melakukan pembangunan dan pembuatan aplikasi dengan cara melakukan pengkodean (*Coding*) sesuai dengan desain yang direncanakan sebelumnya.

#### 3.4.1. Coding

*Coding* dilakukan dengan menggunakan *Visual Studio Code*, XAMPP, CodeIgniter 4, PHP, HTML, CSS, JS, dan MySQL. Pada tahap *coding* juga dilakukan penerapan metode pengendalian persediaan *Min Max*.

#### 3.4.2. Testing

*Testing* dilakukan dengan menggunakan *Black Box Testing*, dan juga dengan menggunakan metode *User Acceptance Testing* (UAT). *Testing* yang pertama dilakukan adalah *Black Box Testing*, pada tahap ini aplikasi diuji oleh *end user*

dimulai dari *input*, proses, dan *output* dari aplikasi apakah sudah tidak ada *error* dari aplikasi. Setelah dilakukan *Black Box Testing* dilanjutkan ke *User Acceptance Testing* (UAT). Proses UAT dilakukan dengan dijalankan aplikasi oleh *user* kemudian diberikan kuisioner kepada *user* mengenai fungsi aplikasi apakah sudah sesuai atau tidak. Skenario untuk *black box* testing dapat dilihat pada Tabel 3.1 sampai Tabel 3.3, sedangkan untuk scenario testing UAT dapat dilihat pada Tabel 3.4 dan Tabel 3.5.

Tabel 3.1 Skenario *Black Box Testing Login*

No	Skenario
<b>Login (Memastikan pengguna sudah terdaftar)</b>	
1	Memasukkan <i>username</i> benar dan <i>password</i> benar
2	Memasukkan <i>username</i> salah dan <i>password</i> benar
3	Memasukkan <i>username</i> salah dan <i>password</i> salah
4	Memasukkan <i>username</i> benar dan <i>password</i> salah

Tabel 3.2 Skenario *Black Box Testing (SA)*

No	Skenario
<b>Permintaan Spare Part (Memastikan pengguna dapat mengelola permintaan spare part)</b>	
1	Memasukkan data permintaan <i>spare part</i> .
2	Tidak memasukkan semua data permintaan.
3	Ubah data permintaan
4	Batalkan permintaan
<b>Melihat Data Spare Part (Memastikan pengguna dapat melihat data spare part)</b>	
5	Melihat data <i>spare part</i>

Tabel 3.3 Skenario *Black Box Testing (Administrasi)*

No	Skenario
<b>Pengelolaan Data Master (Memastikan pengguna dapat mengelola data master)</b>	
1	<i>Input</i> data <i>supplier</i>
2	<i>Input</i> data <i>supplier</i> nama kosong
3	<i>Edit</i> data <i>supplier</i>
4	Hapus data <i>supplier</i>
5	<i>Input</i> data mobil
6	<i>Input</i> data mobil tidak lengkap
7	<i>Edit</i> data mobil
8	Hapus data mobil
9	<i>Input</i> data <i>spare part</i>
10	<i>Input</i> data <i>spare part</i> tidak lengkap
11	<i>Edit</i> data <i>spare part</i>
<b>Spare Part Keluar (Memastikan pengguna dapat mengelola data spare part keluar)</b>	
12	Konfirmasi <i>spare part</i> keluar
13	Stok mendekati atau kurang dari nilai minimum
14	Stok kurang dari permintaan
15	Lihat data <i>spare part</i> keluar
<b>Pemesanan Spare Part (Memastikan pengguna dapat mengelola data pemesanan spare part)</b>	

No	Skenario
16	Memasukkan data pemesanan <i>spare part</i> .
17	Memasukkan data pemesanan <i>spare part</i> dengan nomor permintaan
18	Tidak memasukkan semua data pemesanan <i>spare part</i> .
19	Ubah data pemesanan <i>spare part</i>
20	Hapus data pemesanan <i>spare part</i>
21	<i>Upload file</i> pemesanan, setelah <i>spare part</i> diterima
<b>Penerimaan Spare Part (Memastikan pengguna dapat mengelola data penerimaan <i>spare part</i>)</b>	
22	Konfirmasi penerimaan <i>spare part</i> jumlah terima sesuai dengan jumlah pesan
23	Konfirmasi penerimaan <i>spare part</i> jumlah terima tidak sesuai jumlah pesan
24	<i>Update</i> jumlah penerimaan yang hanya menerima sebagian stok
25	Lihat data penerimaan <i>spare part</i>
<b>Retur Spare Part (Memastikan pengguna dapat mengelola data retur <i>spare part</i>)</b>	
26	Konfirmasi retur <i>spare part</i>
27	Lihat data retur <i>spare part</i>
28	<i>Upload file</i> retur
<b>Melihat Log Spare Part (Memastikan pengguna dapat melihat log <i>spare part</i>)</b>	
29	Lihat log <i>spare part</i>
<b>Laporan (Memastikan pengguna dapat memfilter dan mencetak laporan)</b>	
30	Memfilter laporan <i>spare part</i> keluar
31	<i>Print</i> laporan <i>spare part</i> keluar
32	Memfilter laporan pemesanan <i>spare part</i>
33	<i>Print</i> laporan pemesanan <i>spare part</i>
34	Memfilter laporan penerimaan <i>spare part</i>
35	<i>Print</i> laporan penerimaan <i>spare part</i>
36	Memfilter laporan retur <i>spare part</i>
37	<i>Print</i> laporan retur <i>spare part</i>
<b>Melihat Hasil Perhitungan Min Max (Memastikan pengguna dapat melihat hasil perhitungan <i>min max</i>)</b>	
38	Melihat hasil perhitungan <i>min max</i>
39	Mengetahui daftar dan jumlah <i>spare part</i> yang stoknya mendekati nilai minimum

Tabel 3.4 Skenario UAT (SA)

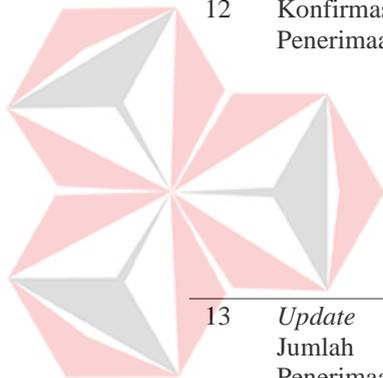
No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
1	<i>Login</i>	Pengguna mencoba untuk <i>login</i> ke dalam aplikasi, apabila berhasil pengguna akan diarahkan ke <i>dashboard</i> jika tidak maka akan muncul tulisan bahwa “ <i>username</i> atau <i>password</i> salah”.			
2	<i>Dashboard</i>	Pengguna dapat melihat jumlah dan daftar permintaan <i>spare part</i> serta <i>pie chart</i> status permintaan dan alasan pembatalan permintaan.			
3	Melihat Permintaan	Pengguna dapat melihat data permintaan.			
4	Menambah Permintaan	Pengguna dapat menambahkan data permintaan, apabila berhasil maka muncul tulisan “data berhasil ditambahkan”.			

5	Mengubah Permintaan	Pengguna dapat mengubah data permintaan, apabila berhasil maka muncul tulisan “data berhasil diubah”.
6	Membatalkan Permintaan	Pengguna dapat membatalkan permintaan dengan memilih alasan pembatalan apabila berhasil maka muncul tulisan “permintaan berhasil dibatalkan”.
7	Melihat <i>Spare Part</i>	Pengguna dapat melihat data <i>spare part</i> .

Tabel 3.5 Skenario UAT (Administrasi)

No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
1	<i>Login</i>	Pengguna melakukan <i>login</i> ke dalam aplikasi, apabila berhasil pengguna akan diarahkan ke <i>dashboard</i> jika tidak maka akan muncul tulisan bahwa “ <i>username</i> atau <i>password</i> salah”.			
2	<i>Dashboard</i>	Pengguna dapat melihat jumlah dan daftar permintaan, <i>spare part</i> keluar, pemesanan, penerimaan, dan retur serta <i>pie chart</i> status permintaan, alasan pembatalan permintaan, status pemesanan, dan status retur.			
3	Pengendalian Persediaan	Pengguna dapat melihat jumlah dan daftar perhitungan <i>min max</i> serta jumlah dan daftar <i>spare part</i> yang harus .			
4	Melihat Permintaan	Pengguna dapat melihat data permintaan.			
5	Konfirmasi <i>Spare Part</i> Keluar	Pengguna dapat mengkonfirmasi <i>spare part</i> keluar, apabila berhasil maka muncul tulisan “data berhasil ditambahkan”.			
6	Melihat <i>Spare Part</i> Keluar	Pengguna dapat melihat data <i>spare part</i> keluar.			
7	Menambah Pemesanan	Pengguna dapat menambahkan pemesanan baik dengan nomor permintaan maupun tanpa nomor permintaan, apabila berhasil maka muncul tulisan “data berhasil ditambahkan”. Jika dicantumkan nomor permintaan maka status permintaan tersebut akan menjadi “permintaan terproses”.			
8	Mengubah Pemesanan	Pengguna dapat mengubah data pemesanan dengan kondisi status pemesanan masih “belum terproses”, apabila berhasil maka muncul tulisan “data berhasil			

No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
		diubah". Jika awalnya tercantun nomor permintaan dan diubah dengan tidak mencantumkan nomor permintaan maka status permintaan tersebut Kembali menjadi permintaan diterima.			
9	Menhapus Pemesanan	Pengguna dapat menghapus pemesanan dengan kondisi status pemesanan masih "belum terproses", apabila berhasil maka akan muncul tulisan "data pemesanan <i>spare part</i> berhasil dihapus".			
10	Upload File Pemesanan	Pengguna dapat mengupload file pemesanan dengan kondisi status pemesanan "diterima". Apabila berhasil akan muncul tulisan "file berhasil diupload".			
11	Melihat Penerimaan	Pengguna dapat melihat data penerimaan.			
12	Konfirmasi Penerimaan	Pengguna dapat konfirmasi penerimaan jika salah satu <i>spare part</i> jumlah penerimaannya kurang dari jumlah pesan maka status penerimaan menjadi "beberapa diterima", sedangkan jika jumlah sama maka status menjadi "diterima semua" apabila berhasil akan muncul tulisan "data berhasil ditambahkan".			
13	Update Jumlah Penerimaan	Pengguna dapat update jumlah penerimaan dengan kondisi status penerimaan "beberapa diterima". Jika berhasil maka muncul tulisan "data berhasil diubah"			
14	Melihat Retur	Pengguna dapat melihat retur.			
15	Konfirmasi Retur	Pengguna dapat konfirmasi retur <i>spare part</i> apabila berhasil maka muncul tulisan "data berhasil ditambahkan" dan dari data penerimaan yang dilakukan retur status returnnya menjadi "✓".			
16	Upload File Retur	Pengguna dapat upload file retur. Apabila berhasil akan muncul tulisan "file berhasil diupload".			
17	Melihat Log Spare Part	Pengguna dapat melihat data log pergerakan <i>spare part</i> .			
18	Print Laporan Spare Part Keluar	Pengguna dapat memfilter dan mencetak laporan <i>spare part</i> keluar dengan keluaran berupa file pdf.			
19	Print Laporan Pemesanan Spare Part	Pengguna dapat memfilter dan mencetak laporan pemesanan dengan keluaran berupa file pdf.			



No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
20	Print Laporan Penerimaan Spare Part	Pengguna dapat memfilter dan mencetak laporan penerimaan dengan keluaran berupa <i>file pdf</i> .			
21	Print Laporan Retur Spare Part	Pengguna dapat memfilter dan mencetak laporan retur dengan keluaran berupa <i>file pdf</i> .			
22	Melihat Supplier	Pengguna dapat melihat data <i>supplier</i> .			
23	Menambah Supplier	Pengguna dapat menambahkan data <i>supplier</i> , apabila berhasil akan muncul tulisan “data berhasil ditambahkan”.			
24	Mengubah Supplier	Pengguna dapat mengubah data <i>supplier</i> , apabila berhasil maka akan muncul tulisan “data berhasil diubah”.			
25	Menghapus Supplier	Pengguna dapat menghapus data <i>supplier</i> dengan kondisi data tersebut tidak terhubung dengan data lain, apabila berhasil muncul tulisan “data <i>supplier</i> berhasil dihapus”.			
26	Melihat Mobil	Pengguna dapat melihat data mobil.			
27	Menambah Mobil	Pengguna dapat menambahkan data mobil, apabila berhasil akan muncul tulisan “data berhasil ditambahkan”.			
28	Mengubah Mobil	Pengguna dapat mengubah data mobil, apabila berhasil maka akan muncul tulisan “data berhasil diubah”.			
29	Menghapus Mobil	Pengguna dapat menghapus data mobil dengan kondisi data tersebut tidak terhubung dengan data lain, apabila berhasil muncul tulisan “data mobil berhasil dihapus”.			
30	Melihat Spare Part	Pengguna dapat melihat data <i>spare part</i> .			
31	Menambah Spare Part	Pengguna dapat menambahkan data <i>spare part</i> , apabila berhasil akan muncul tulisan “data berhasil ditambahkan”.			
32	Mengubah Spare Part	Pengguna dapat mengubah data <i>spare part</i> , apabila berhasil maka akan muncul tulisan “data berhasil diubah”.			
33	Menghapus Spare Part	Pengguna dapat menghapus data <i>spare part</i> dengan kondisi data tersebut tidak terhubung dengan data lain, apabila berhasil muncul tulisan “data <i>spare part</i> berhasil dihapus”.			

## BAB IV HASIL PEMBAHASAN

### 4.1. Hasil *Requirement Gathering*

Berikut merupakan hasil dari *requirement gathering* yang dilakukan pada bengkel Modifie.

#### 4.1.1. Hasil Analisis Proses Bisnis

Alur proses bisnis bagian persediaan pada bengkel Modifie saat ini masih dikerjakan oleh bagian Administrasi dan *Service Advisor*, yaitu dari pihak *Service Advisor* mencatat kebutuhan *spare part* yang dibutuhkan, kemudian *Service Advisor* memberikan laporan list *spare part* yang dibutuhkan kepada pihak Administrasi untuk mengecek ketersediaan *spare part* dan mengambil *spare part* yang dibutuhkan (jika *spare part* tidak ada maka pihak Administrasi akan memesan *spare part* kepada *supplier* melalui *Whatsapp* atau *Gmail*. Setelah dipesan maka pihak *supplier* akan mengirim *spare part*, pihak Administrasi akan menerima dan mengecek *spare part* (jika *spare part* tidak sesuai maka dilakukan proses retur oleh pihak Administrasi, yang nantinya akan diterima oleh *supplier* kemudian *supplier* akan memproses retur tersebut, setelah itu *supplier* akan mengirim *spare part* yang sesuai ke bagian Administrasi) jika sesuai bagian administrasi akan mencatat *spare part* masuk), jika *spare part* tersedia maka akan langsung dilakukan pencatatan *spare part* keluar oleh pihak Administrasi. Untuk proses bisnis saat ini dapat dilihat pada Lampiran 2. Terdapat beberapa masalah dari proses bisnis pada Lampiran 2, yaitu sebagai berikut :

1. Pertama masalah dalam efisiensi pada proses memberikan list *spare part* yang dibutuhkan, pada proses ini terkesan tidak efisien dan membuang waktu dimana SA harus menyerahkan list *spare part* ke Administrasi. (Membutuhkan waktu lama karena beda lantai)
2. Tidak ada kartu stok, sehingga bagian Administrasi harus mengecek ketersediaan *spare part* secara satu-persatu digudang.

3. Sulit melakukan pengecekan barang yang datang dan yang dipesan, karena pemesanan belum terarsip dengan baik.
4. Belum ada metode pengendalian persediaan, yang membuat permintaan pelanggan tidak terpenuhi karena stok *spare part* habis.

Dari masalah diatas bisa diberikan solusi aplikasi berbasis *web* yang memiliki fitur :

1. Pencatatan transaksi, yang nantinya SA dapat memasukkan *list spare part* yang dibutuhkan, agar bagian Administrasi dapat melihat catatan *list spare part* secara *real time*.
2. Memberikan informasi kepada bagian Administrasi mengenai stok *spare part* yang ada.
3. Dapat membuat pemesanan dan mengisikan barang yang diterima, agar dapat lebih mudah mengecek kesesuaian antara *spare part* yang dipesan dan diterima.
4. Pengendalian persediaan menggunakan metode *Min Max* untuk mengurangi kemungkinan kehabisan *spare part*, dan membuat permintaan pelanggan tidak terpenuhi.

Dilihat dari analisis proses bisnis saat ini, masalah yang masih ada pada bengkel Modifie bisa diatasi dengan menggunakan aplikasi pengendalian persediaan menggunakan metode *Min Max*.

#### 1. Hasil Identifikasi Masalah

Identifikasi masalah yang didapatkan dari hasil analisis proses bisnis dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Masalah

Permasalahan	Dampak	Solusi
Permintaan pelanggan yang tidak terpenuhi sekitar 10% karena kehabisan <i>spare part</i> .	Apabila terjadi masalah kehabisan stok atau <i>out of stock</i> akan merugikan bengkel Modifie dari sisi keuntungan dan juga kepuasan pelanggan.	Aplikasi ini tersedia persediaan <i>spare part</i> dengan menggunakan metode <i>Min Max</i> untuk menjaga stok <i>spare part</i> di gudang agar optimal.
Pertama masalah dalam efisiensi pada proses memberikan <i>list spare part</i> yang dibutuhkan, pada	Membuat proses bisnis menjadi lama, karena proses ini memakan waktu.	Aplikasi dengan fitur pencatatan transaksi, yang nantinya SA dapat memasukkan <i>list spare part</i>

Permasalahan	Dampak	Solusi
proses ini terkesan tidak efisien dan membuang waktu dimana SA harus menyerahkan list <i>spare part</i> ke Administrasi. (Membutuhkan waktu lama karena beda lantai)		yang dibutuhkan, dan bagian Administrasi dapat melihat catatan list <i>spare part</i> secara <i>real time</i> .
Tidak ada kartu stok, sehingga bagian Administrasi harus mengecek ketersediaan <i>spare part</i> secara satu-persatu digudang.	Karena proses pengecekan ketersediaan dan daftar <i>spare part</i> harus dicari dan di cek satu persatu, membuat pelanggan menunggu lama.	Aplikasi ini memberikan informasi kepada bagian Administrasi mengenai stok <i>spare part</i> yang ada.
Sulit melakukan pengecekan barang yang datang dan yang dipesan, karena pemesanan belum tersip dengan baik.	Karena banyaknya data transaksi pemesanan pastinya saja memerlukan waktu banyak untuk mencari, dan memfilter, data pemesanan dan bisa terdapat kemungkinan salah dalam mengecek data karena banyaknya data.	Dengan aplikasi ini nantinya dapat membuat pemesanan dan mengisikan barang yang diterima, agar dapat lebih mudah mengecek kesesuaian antara <i>spare part</i> yang dipesan dan diterima.

## 2. Hasil Identifikasi Pengguna

Didapatkan hasil identifikasi pengguna dari wawancara, observasi, dan analisis proses bisnis. Berikut merupakan tugas tanggung jawab setiap karyawan yang berhubungan dengan bagian persediaan (*inventory*).

### a. *Service Advisor* (SA)

1. Mengecek mobil pelanggan
2. Mencatat data *spare part* yang dibutuhkan

### b. Administrasi

1. Menerima list *spare part* yang dibutuhkan dari SA
2. Mengecek ketersediaan *spare part*
3. Memesan *spare part* (PO)
4. Mencatat keluar masuknya *spare part*

## 3. Hasil Identifikasi Kebutuhan Fungsional

Berikut adalah hasil identifikasi kebutuhan fungsional.

- a. *Login*
- b. Pengelolaan data *master*
- c. Mencatat penerimaan *spare part*
- d. Mencatat *spare part* keluar

- e. Mencatat permintaan *spare part*
- f. Melihat data *spare part*
- g. Mencatat pemesanan *spare part*
- h. Mencatat retur *spare part*
- i. Pengendalian persediaan (metode *min max*)
- j. Mencatat perubahan stok *spare part*
- k. Melihat *log spare part*
- l. Mencetak laporan

#### 4. Hasil Identifikasi Data

Berikut merupakan hasil identifikasi data yang diperlukan yaitu: data *user*, *spare part*, *supplier*, mobil, jumlah *spare part* rusak/tidak sesuai, permintaan, detail permintaan, *spare part* keluar, detail *spare part* keluar, pemesanan, detail pemesanan, penerimaan, detail penerimaan, retur, detail retur, dan *log spare part*.

##### 4.1.2. Hasil Kebutuhan Pengguna

Didapat dari hasil identifikasi pengguna, terdapat 2 aktor dalam proses bisnis bagian *inventory* yaitu *Service Advisor* (SA) dan Administrasi. Dari tugas dan tanggung jawab diatas maka didapatkan kebutuhan pengguna dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Kebutuhan Pengguna *Service Advisor* (SA)

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
<i>Login</i>	Data <i>user</i>	Validasi <i>username</i> dan <i>password</i>
Mencatat permintaan <i>spare part</i>	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i></li> <li>2. Data mobil</li> <li>3. Data permintaan</li> <li>4. Data detail permintaan</li> </ol>	<ol style="list-style-type: none"> <li>1. <i>Spare part</i> yang dibutuhkan</li> <li>2. Daftar mobil</li> <li>3. Permintaan pelanggan</li> </ol>
Melihat data <i>spare part</i>	Data <i>spare part</i>	<ol style="list-style-type: none"> <li>1. Jumlah <i>spare part</i></li> <li>2. Daftar <i>spare part</i></li> </ol>

Tabel 4.3 Kebutuhan Pengguna Bagian Administrasi

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
<i>Login</i>	Data <i>user</i>	Validasi <i>username</i> dan <i>password</i>
Pengelolaan data <i>master</i>	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i></li> <li>2. Data <i>supplier</i></li> <li>3. Data mobil</li> </ol>	<ol style="list-style-type: none"> <li>1. Daftar <i>spare part</i></li> <li>2. Daftar <i>supplier</i></li> <li>3. Daftar mobil</li> </ol>

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
Mencatat penerimaan <i>spare part</i>	<ol style="list-style-type: none"> <li>1. Data <i>supplier</i></li> <li>2. Data <i>spare part</i></li> <li>3. Data pemesanan</li> <li>4. Data detail pemesanan</li> <li>5. Data penerimaan</li> <li>6. Data detail penerimaan</li> </ol>	<ol style="list-style-type: none"> <li>1. Daftar dan jumlah <i>spare part</i> yang diterima</li> <li>2. Daftar <i>supplier</i> yang mengirim <i>spare part</i></li> <li>3. Daftar dan jumlah <i>spare part</i> yang dipesan</li> </ol>
Mencatat <i>spare part</i> keluar	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i></li> <li>2. Data permintaan</li> <li>3. Data detail permintaan</li> <li>4. Data <i>spare part</i> keluar</li> <li>5. Data detail <i>spare part</i> keluar</li> </ol>	<ol style="list-style-type: none"> <li>1. Jumlah dan daftar <i>spare part</i> yang keluar</li> <li>2. Jumlah dan daftar permintaan <i>spare part</i></li> </ol>
Mencatat pemesanan <i>spare part</i>	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i></li> <li>2. Data <i>supplier</i></li> <li>3. Data pemesanan</li> <li>4. Data detail pemesanan</li> </ol>	<ol style="list-style-type: none"> <li>1. Jumlah <i>spare part</i> yang dipesan</li> <li>2. Daftar <i>supplier</i> yang mengakomodir pesanan <i>spare part</i></li> </ol>
Mencatat retur <i>spare part</i>	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i></li> <li>2. Data <i>supplier</i></li> <li>3. Data penerimaan</li> <li>4. Data detail penerimaan</li> <li>5. Data <i>spare part</i> rusak/tidak sesuai</li> </ol>	<ol style="list-style-type: none"> <li>1. Daftar <i>spare part</i> yang diretur</li> <li>2. Jumlah <i>spare part</i> yang diretur</li> <li>3. Daftar <i>supplier</i> yang dimintai retur</li> </ol>
Melihat log <i>spare part</i>	Data log <i>spare part</i>	Informasi pergerakan log <i>spare part</i>
Mencetak laporan	<ol style="list-style-type: none"> <li>1. Data <i>spare part</i> masuk</li> <li>2. Data <i>spare part</i> keluar</li> <li>3. Data pemesanan <i>spare part</i></li> <li>4. Data retur <i>spare part</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Laporan penerimaan</li> <li>2. Laporan <i>spare part</i> keluar</li> <li>3. Laporan pemesanan <i>spare part</i></li> <li>4. Laporan retur <i>spare part</i></li> </ol>

#### 4.1.3. Hasil Kebutuhan Fungsional

Berikut merupakan hasil analisis kebutuhan fungsional yang dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kebutuhan Fungsional

<b>Aktor</b>	<b>Fungsi</b>	<b>Deskripsi</b>
SA dan Administrasi	<i>Login</i>	Merupakan proses untuk <i>user</i> masuk ke dalam sistem.
Administrasi	Pengelolaan data <i>master</i>	Merupakan proses pengelolaan data <i>master</i> yang meliputi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> .
Administrasi	Mencatat penerimaan <i>spare part</i>	Merupakan proses <i>input</i> data <i>spare part</i> masuk dari proses <i>pemesanan</i> .
Administrasi	Mencatat <i>spare part</i> keluar	Merupakan proses dimana dicatatnya <i>spare part</i> keluar baik dari penjualan maupun untuk perbaikan mobil.
SA	Mencatat permintaan <i>spare part</i>	Merupakan proses untuk mencatat permintaan <i>spare part</i> .
SA	Melihat data <i>spare part</i>	Merupakan proses SA untuk bisa melihat data <i>spare part</i> .
Administrasi	Mencatat pemesanan <i>spare part</i>	Merupakan proses pemesanan <i>spare part</i> kepada <i>supplier</i> yang nantinya akan dilakukan pengisian

Aktor	Fungsi	Deskripsi
		<i>form</i> dan data akan dikirim <i>off system</i> melalui <i>gmail supplier</i> untuk dilakukan pemesanan.
Administrasi	Mencatat retur <i>spare part</i>	Merupakan proses untuk pengembalian <i>spare part</i> kepada <i>supplier</i> yang disebabkan oleh barang salah maupun rusak.
Sistem	Pengendalian persediaan (metode <i>min max</i> )	Merupakan fungsi yang digunakan untuk mengendalikan persediaan agar tidak terjadinya <i>overstock</i> maupun <i>out of stock</i> dengan menggunakan metode <i>min max</i>
Sistem	Mencatat perubahan stok <i>spare part</i>	Fungsi untuk mencatat pergerakan <i>log spare part</i> .
Administrasi	Melihat <i>log spare part</i>	Merupakan fungsi untuk melihat data <i>log spare part</i>
Administrasi	Mencetak laporan	Merupakan fungsi yang nantinya digunakan untuk mencetak laporan pada transaksi yang berhubungan dengan <i>spare part</i> .

#### 4.1.4. Hasil Kebutuhan Nonfungsional

Berikut ini merupakan hasil dari analisis kebutuhan nonfungsional.

##### 1. *Availability* (ketersediaan aplikasi)

Aplikasi ini nantinya dapat diakses selama bengkel buka yaitu pada jam kerja.

##### 2. *Portability* (kemudahan)

Aplikasi bisa digunakan pada semua *browser*.

##### 3. *Safety* (keamanan)

Sistem keamanan berupa sistem *login* dengan memasukkan *username* dan *password* agar dapat mengakses aplikasi.

#### 4.1.5. Hasil Kebutuhan Sistem

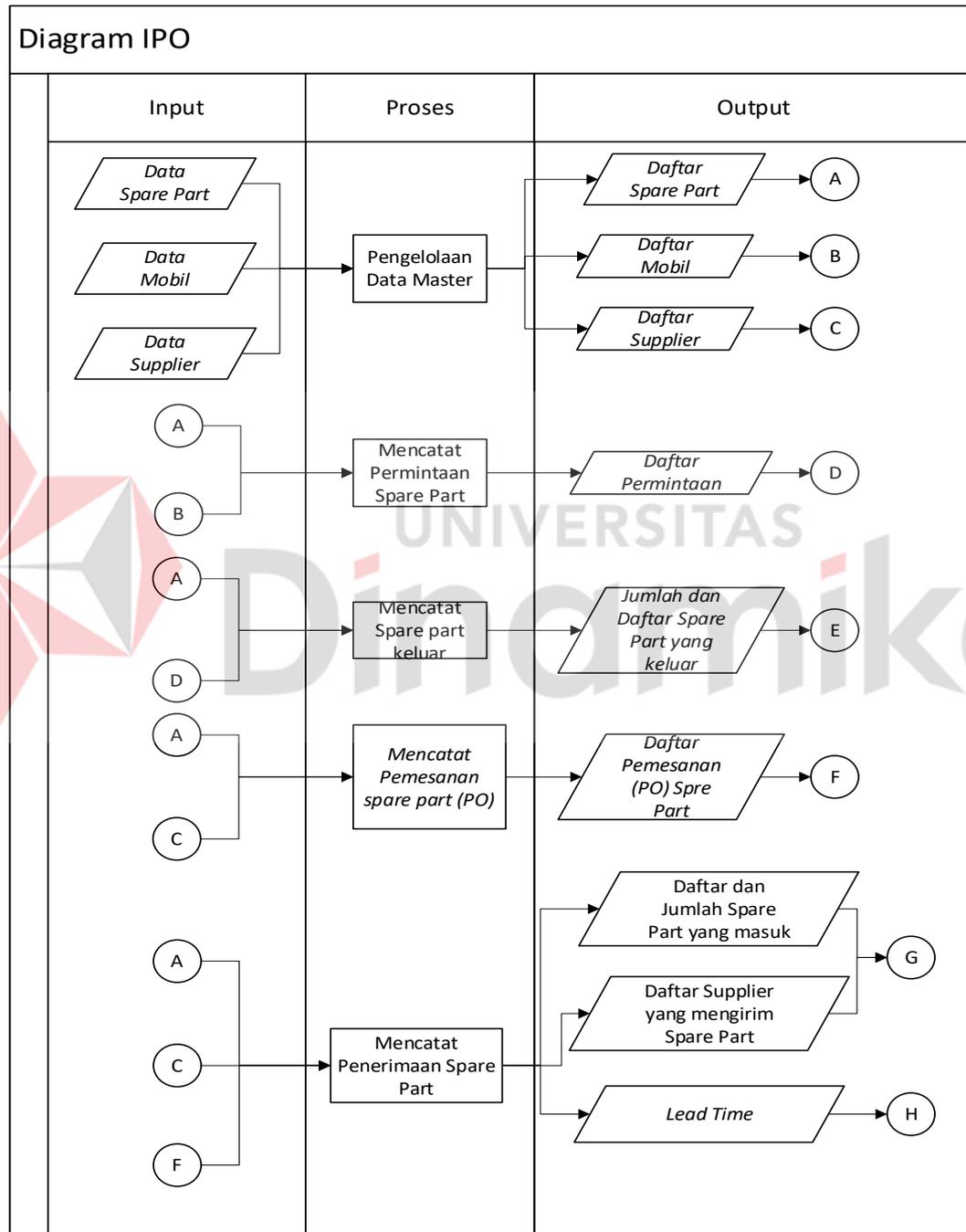
Berikut adalah daftar kebutuhan alat untuk menunjang arsitektur yang dapat dilihat pada Tabel 4.5.

Tabel 4.5 Kebutuhan Sistem

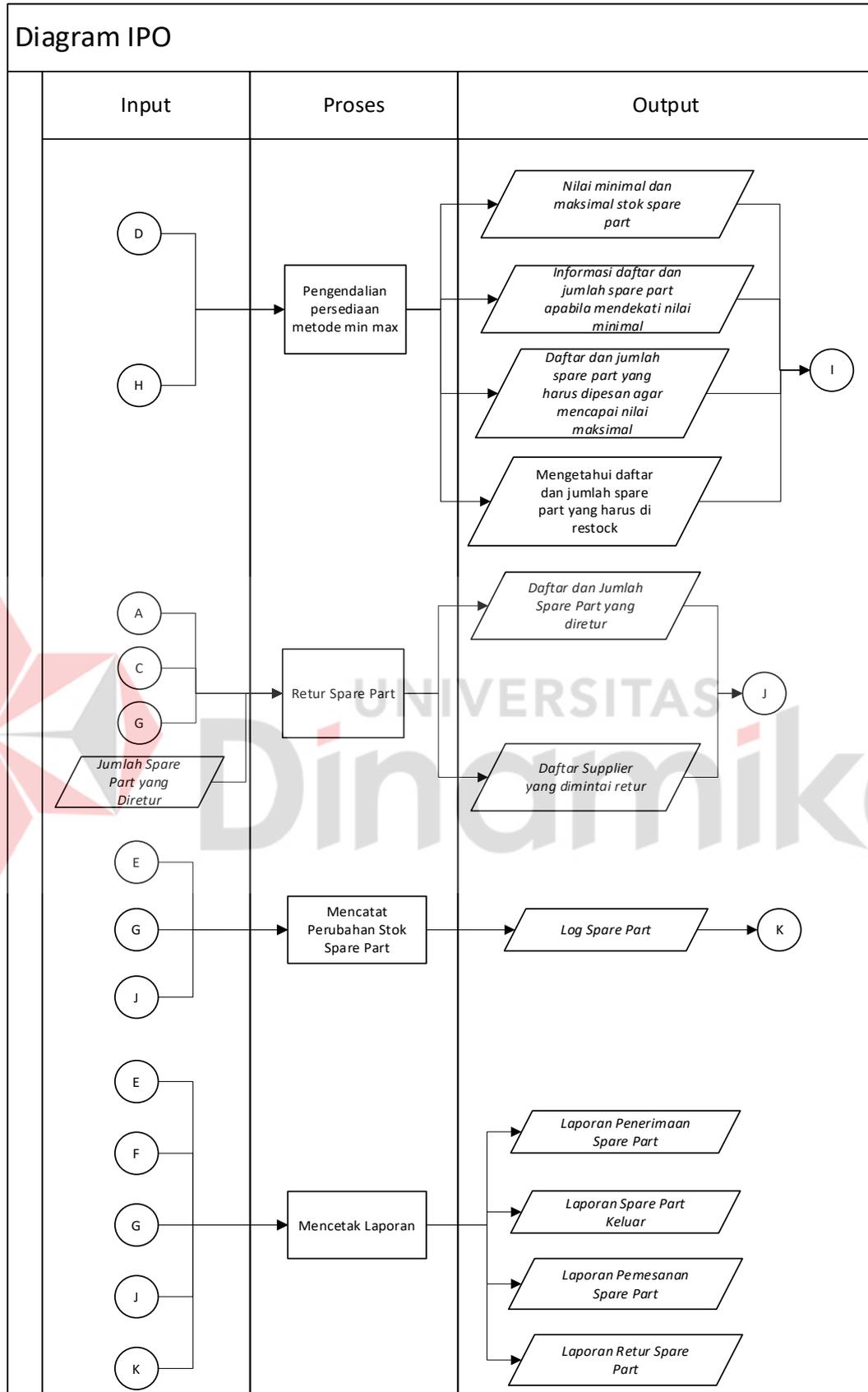
Item	Kelompok	Spesifikasi
<i>Server</i>	<i>Software</i>	Sistem Operasi <i>Windows 7 SP1</i> , 8.1, 10 64 bit, dan <i>Xampp</i> .
Komputer Klien (Pengguna)	<i>Hardware</i>	<i>Processor Intel Core I3</i> atau yang lebih baru, Memori 2Gb atau lebih tinggi, Kapasitas <i>Hardisk</i> 100 Gb atau lebih tinggi.
	<i>Software</i>	Sistem Operasi <i>Windows 7 SP1</i> , 8.1, 10 64 bit, dan <i>web browser</i> .
Komputer Pengembangan	<i>Hardware</i>	<i>Processor Intel Core I3</i> atau yang lebih baru, Memori 2Gb atau yang lebih tinggi, Kapasitas <i>hardisk</i> 150 Gb atau yang lebih tinggi.
	<i>Software</i>	Sistem Operasi <i>Windows 7 SP1</i> , 8.1, 10 64 bit, <i>text editor</i> , <i>XAMPP</i> , dan <i>web browser</i> .

#### 4.1.6. Hasil Diagram IPO

Untuk IPO diagram yang di maksud dapat dilihat pada Gambar 4.1 sampai Gambar 4.2, berikut merupakan penjelasan dari diagram IPO berdasarkan gambar diatas.



Gambar 4.1 Diagram IPO (1)



Gambar 4.2 Diagram IPO (2)

### A. Input

Pada aplikasi ini terdapat beberapa *input* yang akan digunakan, antara lain:

a. Data *spare part*

Data *spare part* yang dimaksudkan berupa kode, nama *spare part*, stok, dan harga.

b. Data mobil

Data mobil yang dimaksudkan berupa merk, tipe, grade, jenis, tahun, *region*, *transmission*, *engine*, dan *frame*.

c. Data *supplier*

Data *supplier* yang dimaksudkan berupa kode, nama *supplier*, alamat *supplier*, dan nomor telepon *supplier*.

d. Jumlah *spare part* yang diretur

Jumlah *spare part* yang diretur merupakan jumlah *spare part* yang rusak ataupun tidak sesuai dari penerimaan *spare part*.

### B. Proses

Berdasarkan *input* yang telah dimasukkan diatas, *input* tersebut akan diproses, dan akan menghasilkan *ouput* atau keluaran yang dibutuhkan. Pada aplikasi ini terdapat beberapa proses antara lain:

a. Pengelolaan data *master*

Pengelolaan data *master* adalah proses pengelolaan data *master* (*spare part*, mobil, dan *supplier*) mulai dari menambah, merubah, dan menghapus data *master* di dalam *database*.

b. Mencatatat permintaan *spare part*

Merupakan proses pencatatan permintaan *spare part* oleh SA. proses pencatatan permintaan *spare part* dimulai dari menambah, merubah, dan membatalkan data didalam *database*, yang nantinya proses ini mengeluarkan data permintaan, data ini akan digunakan untuk proses pengendalian persediaan, pada data permintaan ini digunakan untuk menghitung *safety stock* yaitu permintaan rata-rata dan maksimal, *safety stock* sendiri dibutuhkan untuk perhitungan *min max*.

c. Mencatat *spare part* keluar

Mencatat spare part keluar adalah proses menambahkan data *spare part* yang dikeluarkan dari permintaan *spare part*.

d. Mencatat pemesanan *spare part*

Mencatat pemesanan *spare part* adalah proses pengelolaan data pemesanan *spare part* dimulai dari menambahkan, merubah, dan menghapus data didalam *database*.

e. Mencatat penerimaan *spare part*

Mencatat penerimaan *spare part* adalah proses pencatatan data penerimaan *spare part*. Proses dijalankan setelah diterimanya *spare part* dari *supplier* oleh bagian Administrasi. *Output* dari proses ini juga nantinya akan digunakan untuk proses pengendalian persediaan (*lead time*)

f. Mencatat retur *spare part*

Merupakan proses pencatatan data retur *spare part* yang dilakukan saat adanya *spare part* rusak atau tidak sesuai dari penerimaan *spare part*.

g. Pengendalian persediaan metode *min max*

Pengendalian persediaan metode *min max* merupakan proses untuk mengidentifikasi nilai minimum dan maksimum stok *spare part* yang dimana nanti dibutuhkan data permintaan dan juga *lead time*. Pertama akan diambil data permintaan 30 hari (bulan sebelumnya) untuk didapatkan permintaan maksimum dan rata-rata, kemudian diambil juga data rata-rata pemesanan dan penerimaan untuk didapatkan selisih hari (*lead time*). Contoh untuk perhitungan metode *Min Max* adalah sebagai berikut, didapatkan data permintaan pada *spare part* A yaitu permintaan maksimum adalah 70, permintaan rata-rata adalah 45, dan *lead time* 5 hari dalam sebulan (Data dibawah adalah contoh):

1. Pertama harus menentukan *safety stock*

$$SS = (\text{Permintaan maksimum} - K) \times W$$

$$SS = (70 - 45) \times 5$$

$$SS = 25 \times 5$$

$$SS = 125$$

2. Kedua menghitung *Minimum* stok

$$Min = (K \times W) + SS$$

$$Min = (45 \times 5) + 125$$

$$Min = 350$$

3. Ketiga menghitung *Maximum* stok

$$Max = 2(K \times W)$$

$$Max = 2(45 \times 5)$$

$$Max = 450$$

4. Terakhir adalah menentukan Q

$$Q = Max - Min$$

$$Q = 450 - 350$$

$$Q = 100$$

- h. Mencatat perubahan *stok spare part*

Merupakan proses untuk mencatat perubahan *stok spare part*, proses ini berjalan otomatis dimana saat dilakukan proses transaksi yang memengaruhi stok seperti *spare part* keluar, penerimaan, dan retur.

- i. Mencetak laporan

Merupakan proses pencetakan laporan mengenai *spare part* yang ada (*spare part* keluar, pemesanan, penerimaan, dan retur *spare part*), proses dilakukan dengan menekan cetak laporan sesuai dengan laporan yang diinginkan.

### C. Output

Dari *input* yang telah dimasukkan dan diproses, maka sistem akan menghasilkan keluaran dan informasi sebagai berikut :

- a. Daftar *spare part*

Daftar *spare part* akan menampilkan data *spare part* yang telah tersimpan pada *database*.

- b. Daftar mobil

Daftar mobil akan menampilkan data mobil yang telah tersimpan pada *database*.

- c. Daftar *supplier*

Daftar *supplier* akan menampilkan data *supplier* yang telah tersimpan pada *database*.

- d. Daftar permintaan

Daftar permintaan akan menampilkan data permintaan yang akan disimpan

pada *database*. Nantinya daftar permintaan digunakan untuk melakukan perhitungan *Min Max*.

- e. Jumlah dan daftar *spare part* yang keluar  
Merupakan keluaran informasi berupa jumlah dan daftar *spare part* yang dikeluarkan oleh Administrasi.
- f. Daftar pemesanan *spare part*  
Merupakan keluaran informasi pemesanan *spare part* yang dipesan dari *supplier*.
- g. Daftar *supplier* yang mengirim *spare part*  
Merupakan keluaran informasi berupa *supplier* mana saja yang mengirim *spare part*.
- h. Daftar dan jumlah *spare part* yang diterima  
Merupakan keluaran informasi berupa daftar dan jumlah *spare part* yang diterima.
- i. *Lead Time*  
Merupakan informasi mengenai *lead time* yang nantinya digunakan untuk menghitung pengendalian persediaan menggunakan metode *min max*.
- j. Nilai minimal dan maksimal stok *spare part*  
Merupakan keluaran informasi berupa nilai minimal dan maksimal stok *spare part* yang nantinya berfungsi untuk memunculkan notifikasi stok hampir habis.
- k. Informasi daftar dan jumlah *spare part* apabila mendekati nilai minimal  
Merupakan keluaran informasi berupa informasi yang menandakan bahwa stok *spare part* harus segera *di restock*.
- l. Daftar dan jumlah *spare part* yang harus dipesan agar mencapai nilai maksimal  
Merupakan keluaran informasi berupa jumlah *spare part* yang harus dipesan sesuai dengan stok saat ini yang menipis agar mencapai nilai maksimal.
- m. Mengetahui daftar dan jumlah *spare part* yang harus *di restock*  
Merupakan keluaran untuk bagian Administrasi agar dapat mengetahui daftar dan jumlah *spare part* yang harus *di restock*.
- n. Daftar dan jumlah *spare part* yang diretur  
Merupakan keluaran informasi berupa daftar dan jumlah *spare part* yang

diretur ke *supplier*.

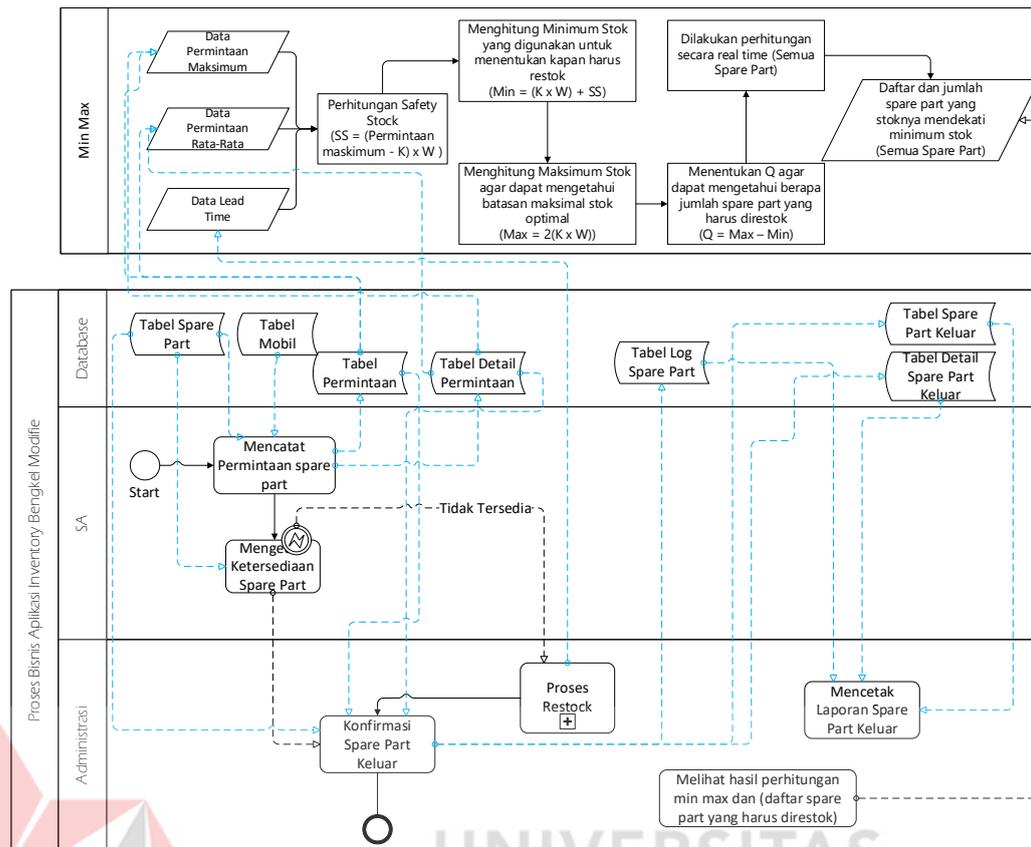
- o. Daftar *supplier* yang dimintai retur  
Merupakan keluaran informasi berupa daftar *supplier* yang dimintai retur.
- p. *Log spare part*  
Merupakan keluaran dari proses otomatis yang terjadi apabila stok *spare part* berubah..
- q. Laporan penerimaan *spare part*  
Laporan penerimaan *spare part* akan mengeluarkan rangkuman dari transaksi *spare part* yang diterima berupa tabel dalam bentuk pdf.
- r. Laporan *spare part* keluar  
Laporan *spare part* keluar akan mengeluarkan rangkuman dari transaksi *spare part* yang keluar berupa tabel dalam bentuk pdf.
- s. Laporan pemesanan *spare part*  
Laporan pemesanan *spare part* akan mengeluarkan rangkuman dari transaksi pemesanan *spare part* berupa tabel dalam bentuk pdf.
- t. Laporan retur *spare part*  
Laporan retur *spare part* akan mengeluarkan rangkuman dari transaksi retur *spare part* berupa tabel dalam bentuk pdf.

## 4.2. Hasil Desain

Berikut merupakan hasil dari desain yang telah dirancang dan dibuat pada bengkel Modifie.

### 4.2.1. Hasil Sysflow

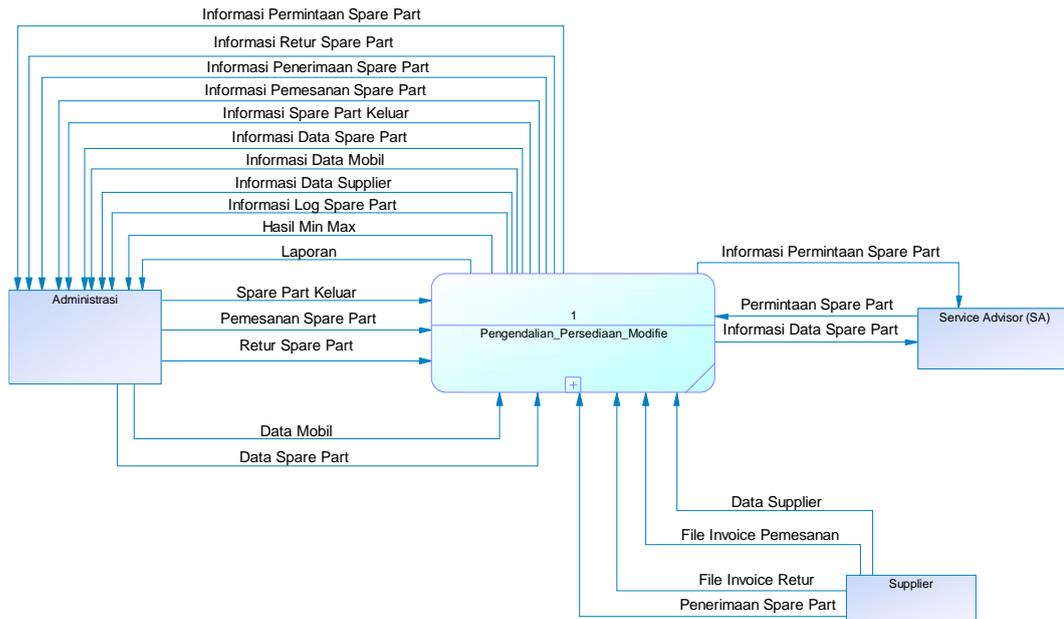
Diasumsikan pada *sysflow* aplikasi, sudah dilakukan pengelolaan data master, pada *sysflow* digunakan 2 garis *flow* berwarna hitam dan biru, untuk *flow* biru berhubungan dengan tabel *database*, sedangkan yang hitam untuk kesesama proses, *sysflow* pada aplikasi yang akan digunakan nantinya dapat dilihat pada Gambar 4.3, sedangkan untuk *sysflow restock* dan retur dapat dilihat pada Lampiran 3.



Gambar 4.3 Sysflow (Utama)

#### 4.2.2. Hasil Data Flow Diagram (DFD)

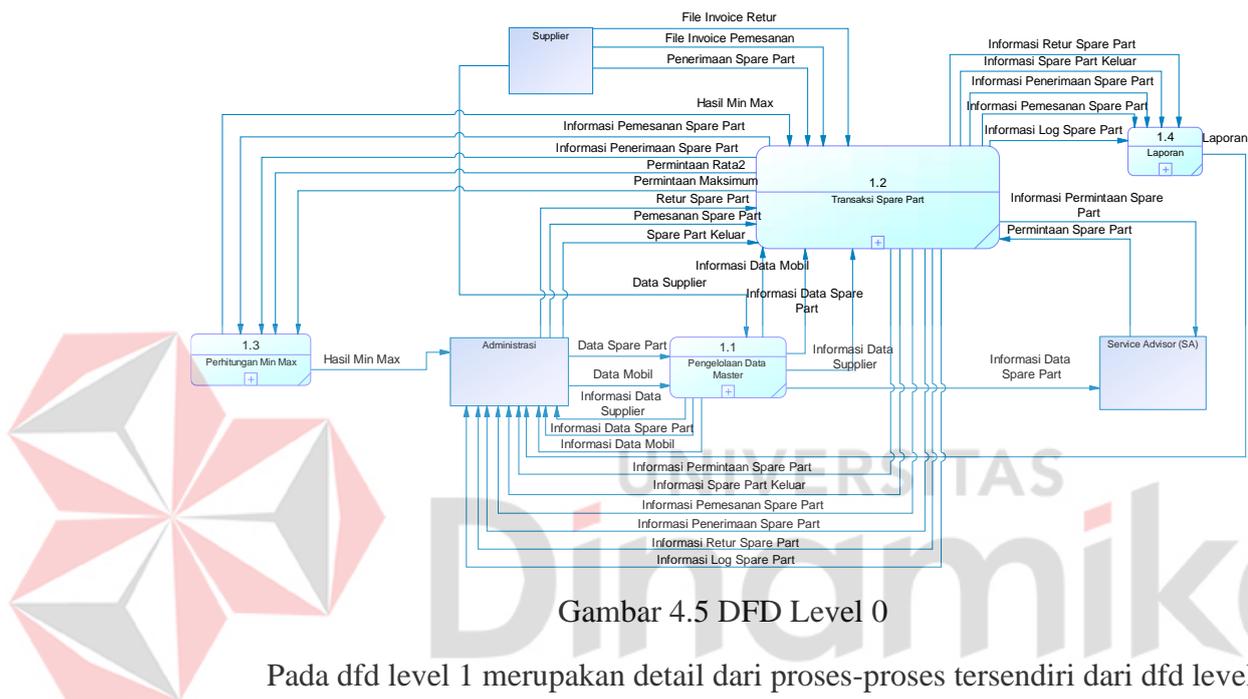
Untuk *data flow diagram* akan dibuat sampai dengan level 1, dengan urutan *context diagram*, dfd level 0, level 1, dan level 2. Pada *context diagram* hanya akan dijelaskan mengenai data yang akan didapat dan dikeluarkan oleh tiap aktor (Administrasi, *Service Advisor*, dan *supplier*) dalam aplikasi nantinya, kemudian akan dijelaskan lebih detail lagi mengenai data yang akan berhubungan dengan melalui proses-proses tersendiri pada level 0, dan seterusnya sampai dengan level 2. Untuk *context diagram* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Context Diagram

Sebelum dfd level 0 dibuat, digambarkan diagram berjenjang yang nantinya berisi mengenai proses-proses yang akan ada di di dfd dari level 0 sampai level 2, diagram berjenjang dapat dilihat pada Lampiran 4. Diagram berjenjang terdapat 4 proses level 0 yaitu pengelolaan data *master*, transaksi *spare part*, perhitungan *min max*, dan laporan, untuk proses pada level 1 terdapat 3 proses dari pengelolaan data master (pengelolaan data *supplier*, mobil, dan *spare part*), 6 proses pada transaksi *spare part* (*spare part* keluar, permintaan, pemesanan, penerimaan, retur, dan *log spare part*) dari proses yang ada pada level 1 akan diturunkan ke level 2 seperti pada pengelolaan data *supplier*, mobil, dan *spare part* dengan proses melihat, *input*, *edit*, dan *delete*. kemudian terdapat proses dari permintaan *spare part*, *spare part* keluar, dan pemesanan *spare part*. Dari ketiga proses permintaan dan pemesanan memiliki proses yang hampir sama kecuali batal, hapus, dan *upload file*, sedangkan untuk *spare part* keluar memiliki 2 proses untuk konfirmasi dan lihat data saja. Proses selanjutnya dari penerimaan *spare part*, retur *spare part*, dan *log spare part*. Penerimaan memiliki proses konfirmasi, update jumlah penerimaan, dan lihat data penerimaan, untuk retur memiliki proses konfirmasi, lihat data, dan *upload file* retur, sedangkan untuk *log spare part* hanya melihat data *log spare part*. Proses perhitungan *min max* memiliki proses untuk menghitung *lead time*, *safety stock*, *min max*, dan untuk melihat daftar dan jumlah *spare part* yang harus dipesan. Untuk proses laporan hanya untuk melakukan *print* atau mencetak laporan dari transaksi

penerimaan *spare part*, *spare part* keluar, retur *spare part*, dan pemesanan *spare part*. Pada dfd level 0 memiliki 4 (empat) proses sesuai dengan diagram berjenjang yang telah di gambarkan, proses dfd level 0 sendiri yaitu terdiri dari pengelolaan data master sampai dengan laporan dan pada dfd level 0 tidak digambarkan tabelnya hanya proses saja dikarenakan proses yang ada terlalu banyak dan akan membuat dfd level 0 sulit untuk dibaca. Untuk lebih detailnya dfd level 0 bisa dilihat pada Gambar 4.5.



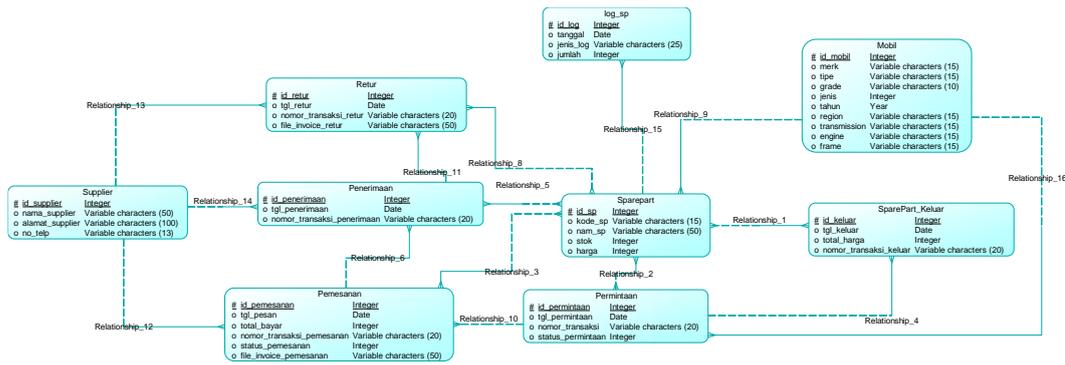
Gambar 4.5 DFD Level 0

Pada dfd level 1 merupakan detail dari proses-proses tersendiri dari dfd level 0 pada Gambar 4.5. Maka dari itu terdapat 4 (empat) dfd level 1. Untuk proses transaksi *spare part* nantinya pada level 1 akan mempunyai 6 proses yaitu, *spare part* keluar, permintaan, pemesanan, penerimaan, retur, dan *log spare part*. Pada proses transaksi *spare part* belum digambarkan tabel dan akan ditampilkan pada dfd level 2 yang akan dibuat nantinya. Pada dfd level 1 pengelolaan data master merupakan proses pengelolaan data pada 3 (tiga) tabel *master* yaitu *supplier*, mobil, dan *spare part*, proses yang diturunkan dari ketiga proses ini memiliki proses yang sama yaitu melihat, *input*, *edit*, dan hapus. Pada dfd level 1 pengelolaan data master berhubungan dengan proses lainnya yaitu transaksi *spare part*, terhubungannya proses-proses tersebut dikarenakan dibutuhkannya informasi dari data master (*supplier*, *spare part*, dan mobil) agar proses tersebut dapat dijalankan. Pada dfd level 2 permintaan *spare part* merupakan proses untuk mencatat dan mengelola data permintaan *spare part* (lihat, *input*, *edit*, dan batalkan permintaan) yang diajukan

oleh *Service Advisor* dan akan menghasilkan data permintaan *spare part* yang dapat dilihat oleh Administrasi. Sedangkan untuk proses *spare part* keluar merupakan proses yang dilakukan oleh Administrasi untuk mencatat data *spare part* keluar (konfirmasi dan lihat *spare part* keluar). Pada proses pemesanan *spare part*, penerimaan *spare part*, dan retur *spare part* hampir sama dengan proses permintaan *spare part* dan *spare part* keluar, hanya berbeda pada proses pemesanan, penerimaan, dan retur berhubungan dengan *supplier*. Pada pemesanan berhubungan dengan penerimaan, sedangkan untuk retur akan berhubungan pada penerimaan. Aktor yang terlibat hanya Administrasi dan *supplier* dalam proses ini untuk proses pada pemesanan adalah lihat, *input*, *edit*, hapus, dan *upload file* pemesanan, untuk penerimaan terdapat 3 proses yaitu lihat, konfirmasi, dan *update* jumlah penerimaan, retur juga memiliki 3 proses yaitu lihat, konfirmasi, dan *upload file* retur. Pada proses laporan sendiri hanya berfokus pada pembuatan laporan dari proses-proses transaksi yang ada (*spare part* keluar, pemesanan *spare part*, penerimaan *spare part*, dan retur *spare part*). Untuk aktor pada proses laporan hanya Administrasi saja, yaitu dengan Administrasi melakukan *print* pada proses transaksi yang akan menghasilkan laporan (pdf). Selain itu dibutuhkan juga *log spare part* untuk laporan *spare part keluar*, penerimaan, dan retur. *Log spare part* sendiri hanya mempunyai 1 proses yaitu melihat *log spare part*. Pada proses perhitungan *min max* terdapat 4 proses yaitu menghitung *lead time*, *safety stock*, menghitung *min max*, dan melihat daftar dan jumlah *spare part* yang harus dipesan. Untuk dfd level 1 sampai 2 dapat dilihat pada Lampiran 5.

#### **4.2.3. Hasil *Conceptual Data Model* (CDM)**

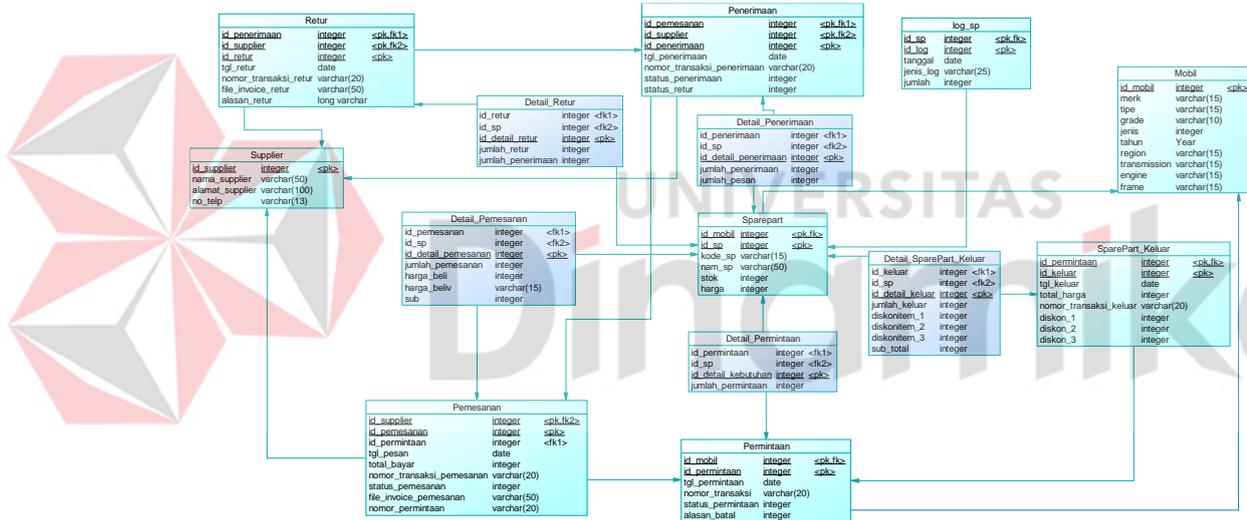
Berikut merupakan *conceptual data model* pada aplikasi nantinya, untuk gambar *conceptual data model* dapat dilihat pada Gambar 4.6.



Gambar 4.6 CDM

4.2.4. Hasil Physical Data Model (PDM)

Berikut merupakan *physical data model* pada aplikasi nantinya, untuk gambar *physical data model* dapat dilihat pada Gambar 4.7.

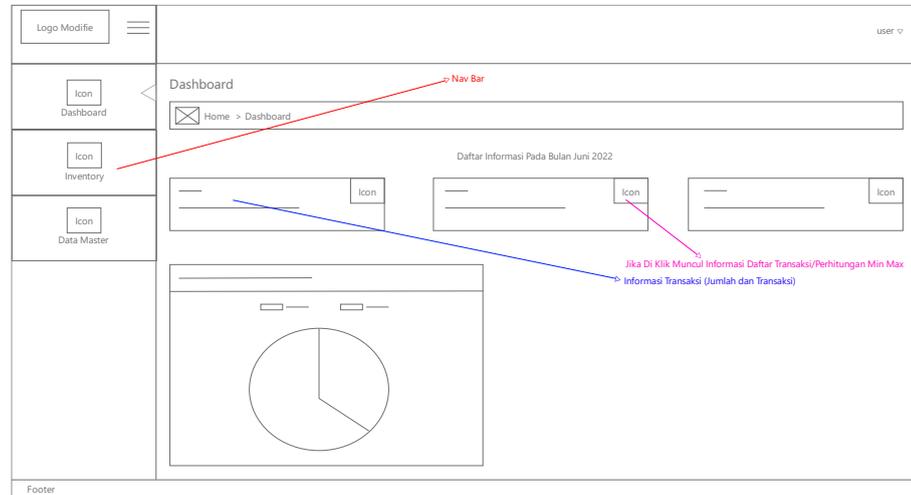


Gambar 4.7 PDM

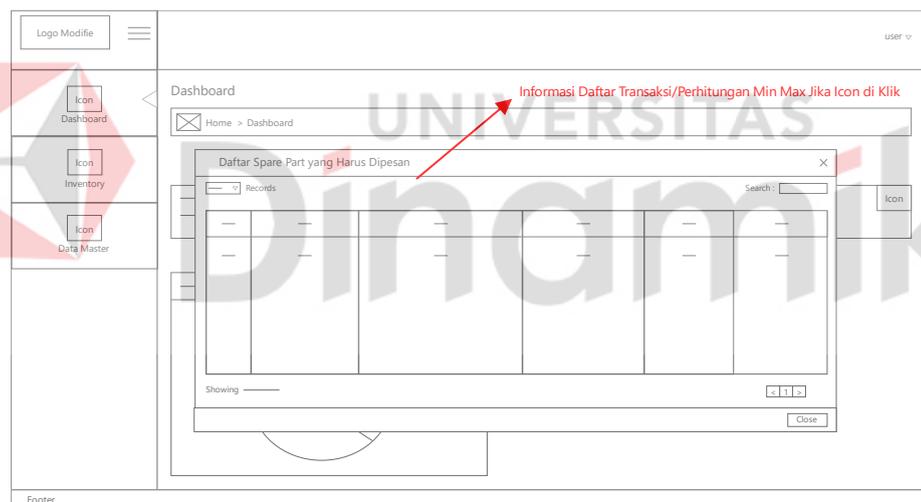
4.2.5. Hasil Desain Interface

Untuk desain *interface* yang dibuat berjumlah 38 desain, dimulai dari *login*, *dashboard*, *form* permintaan, *spare part* keluar, pemesanan, penerimaan, retur, laporan *spare part* keluar, pemesanan, penerimaan, retur, dan juga *form* pengelolaan data master. Desain *interface* pengendalian persediaan akan terdapat pada *dashboard*, pada *dashboard* akan terdapat informasi jumlah dan daftar dari transaksi yang ada (*spare part* keluar, permintaan, pemesanan, penerimaan, dan retur) serta daftar dan jumlah hasil perhitungan *min max* serta *spare part* yang harus

di pesan dengan menekan icon pada pojok kanan atas yang dapat dilihat pada Gambar 4.8 dan Gambar 4.9 sedangkan untuk desain *interface* yang lainnya dapat dilihat pada Lampiran 6.



Gambar 4.8 Desain *Dashboard*



Gambar 4.9 Desain Hasil Perhitungan *Min Max*

### 4.3. Implementasi Sistem

Berikut merupakan implementasi dari aplikasi yang sudah dibuat pada bengkel Modifie.

#### 4.3.1. Implementasi *Service Advisor (SA)*

Implementasi sistem pada menjelaskan terkait fungsi atau fitur yang dapat diakses. Berikut adalah implementasi pada *Service Advisor (SA)*.

### **A. Login**

Pada fungsi ini *user* memasukan *username* dan *password* apabila *username* dan *password* benar maka akan langsung masuk ke *dashboard* jika salah akan muncul *error* peringatan *username* atau *password* salah. Untuk implementasi dapat dilihat pada Lampiran 7.

### **B. Melihat Data Spare Part**

Pada fungsi melihat data *spare part Service Advisor* dapat melihat daftar *spare part*, seperti *spare part* dari mobil apa, kode, nama, harga, dan stok *spare part*. Untuk implementasi dapat dilihat pada Lampiran 7.

### **C. Proses Permintaan Spare Part**

Pada fungsi ini *Service Advisor* dapat melihat data permintaan, menambahkan data permintaan, ubah data permintaan, dan membatalkan permintaan. Untuk implementasi dapat dilihat pada Lampiran 7.

## **4.3.2. Implementasi Administrasi**

Implementasi sistem pada menjelaskan terkait fungsi atau fitur yang dapat diakses Administrasi. Berikut adalah implementasi pada *Service Advisor* (SA).

### **A. Login**

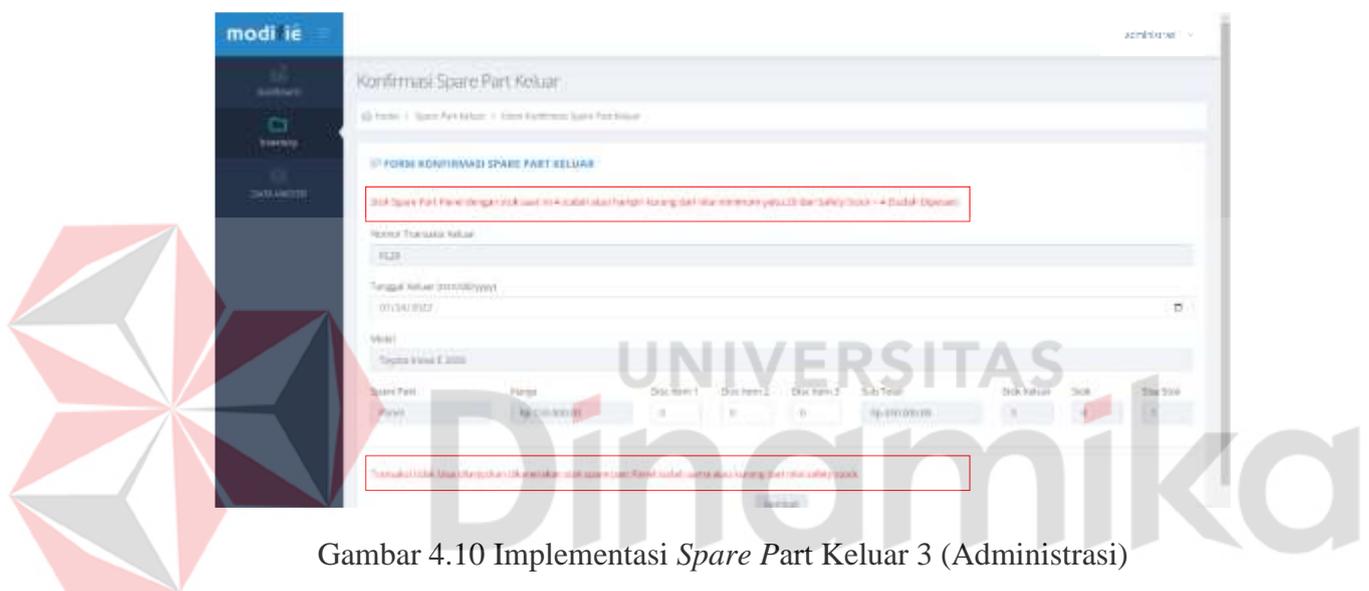
Pada fungsi ini *user* memasukan *username* dan *password* apabila *username* dan *password* benar maka akan langsung masuk ke *dashboard* jika salah akan muncul *error* peringatan *username* atau *password* salah. Untuk implementasi dapat dilihat pada Lampiran 7.

### **B. Pengelolaan Data Master**

Pada fungsi ini *user* melakukan pengelolaan data master mulai dari *input*, *edit*, dan *delete* data. Terdapat 3 data master yaitu *supplier*, mobil, dan *spare part*. Hasil implementasi dapat dilihat pada Lampiran 7.

### C. Spare Part Keluar

Pada fungsi ini *user* melakukan konfirmasi untuk keluarnya *spare part* dan juga melihat data *spare part* keluar. Pada proses *spare part* keluar akan dilakukan pengecekan apabila stok *spare part* yang akan di keluarkan mendekati atau kurang dari nilai *minimum stock* dari hasil perhitungan *min max*, kemudian juga akan di cek apakah sudah dilakukan pemesanan pada *spare part* tersebut atau tidak, dan juga apabila stok *spare part* sama atau kurang dari *safety stock* maka transaksi tidak dapat dilanjutkan, untuk lebih detail dapat dilihat pada Gambar 4.10. Untuk hasil implementasi yang lain dapat dilihat pada Lampiran 7.



Gambar 4.10 Implementasi *Spare Part* Keluar 3 (Administrasi)

### D. Pemesanan Spare Part

Pada fungsi ini *user* membuat pemesanan *spare part*. Untuk hasil implementasi dapat dilihat pada Lampiran 7.

### E. Konfirmasi Penerimaan Spare Part

Pada fungsi ini *user* melakukan konfirmasi penerimaan *spare part*. Untuk hasil implementasi dapat dilihat pada Lampiran 7.

### F. Retur Spare Part

Pada fungsi ini *user* melakukan konfirmasi retur *spare part*. Untuk hasil implementasi dapat dilihat pada Lampiran 7.

### G. Melihat *Log Spare Part*

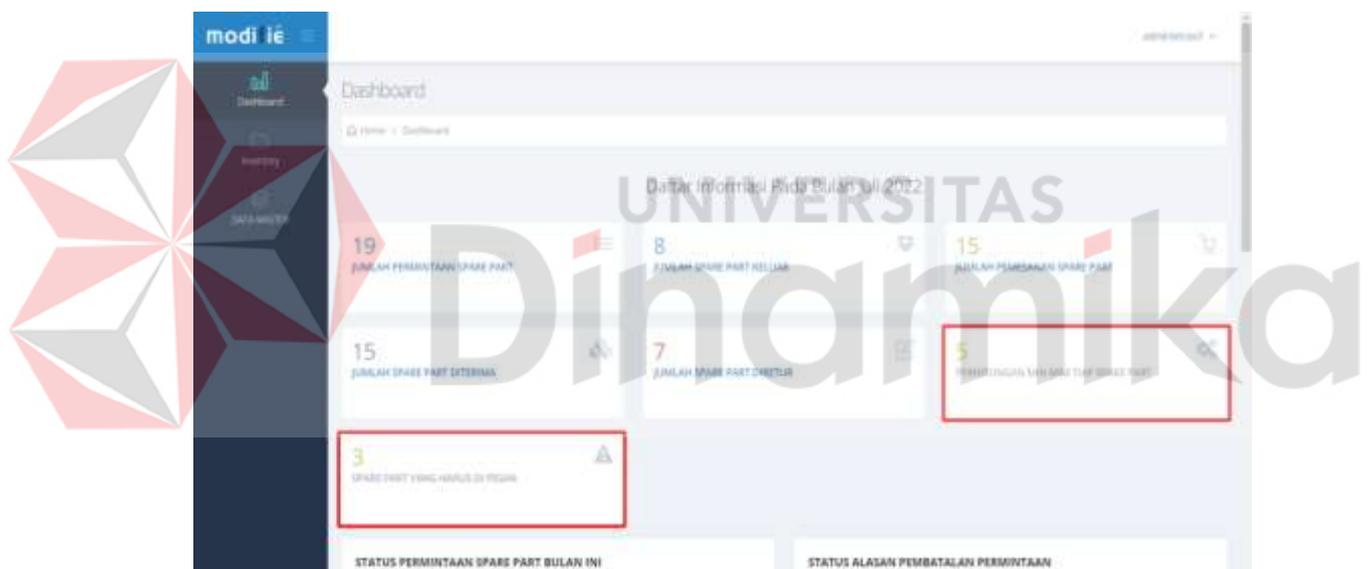
Pada fungsi ini *user* melihat daftar dari pergerakan *log spare part*. Untuk implementasi dapat dilihat pada Lampiran 7.

### H. Laporan

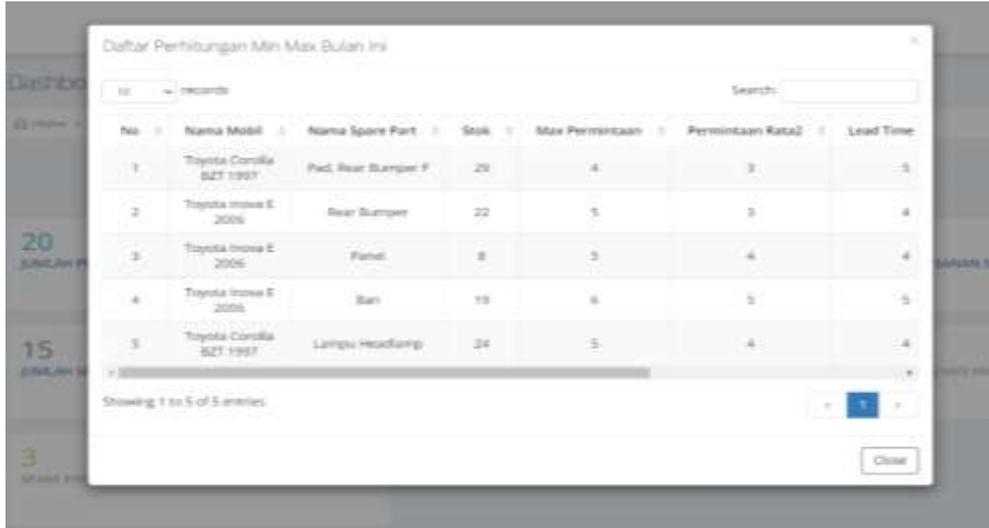
Pada fungsi ini *user* melakukan *print* laporan dari *spare part* keluar sampai retur. Untuk hasil implementasi dapat dilihat pada pada Lampiran 7.

### I. Melihat Hasil Perhitungan *Min Max*

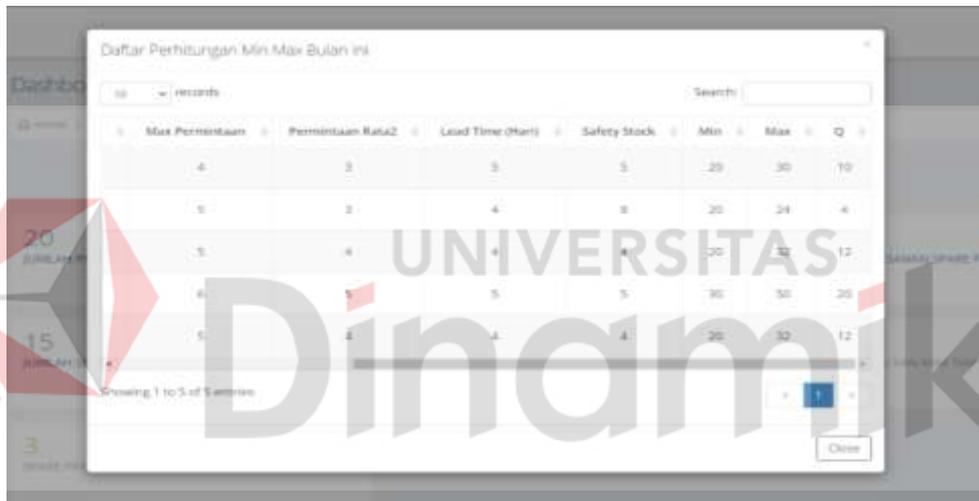
Pada fungsi ini *user* dapat melihat hasil perhitungan *min max*, dan juga daftar dan jumlah *spare part* yang harus di *restock*. Untuk hasil implementasi dapat dilihat pada Gambar 4.11 sampai Gambar 4.16.



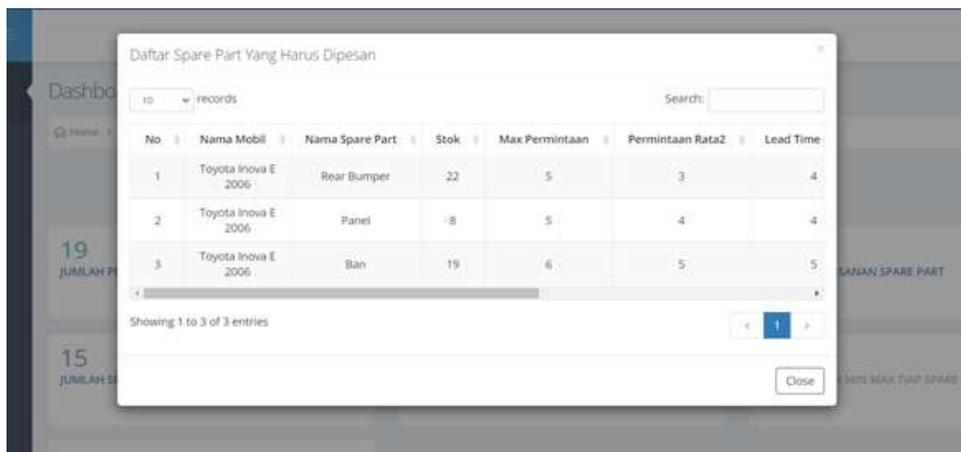
Gambar 4.11 Implementasi Hasil Perhitungan *Min Max* 1



No	Nama Mobil	Nama Spare Part	Stok	Max Permintaan	Permintaan Rata2	Lead Time
1	Toyota Corolla 827 1997	Pod. Rear Bumper F	20	4	3	5
2	Toyota Inova E 2006	Rear Bumper	22	5	3	4
3	Toyota Inova E 2006	Panel	8	3	4	4
4	Toyota Inova E 2006	Ban	19	6	5	5
5	Toyota Corolla 827 1997	Lampu hoodlamp	24	5	4	4

Gambar 4.12 Implementasi Hasil Perhitungan *Min Max 2*


Max Permintaan	Permintaan Rata2	Lead Time (Hari)	Safety Stock	Min	Max	Q
4	3	5	5	20	30	10
5	3	4	8	20	24	4
5	4	4	4	20	30	12
6	5	5	5	30	50	20
5	4	4	4	20	30	12

Gambar 4.13 Implementasi Hasil Perhitungan *Min Max 3*


No	Nama Mobil	Nama Spare Part	Stok	Max Permintaan	Permintaan Rata2	Lead Time
1	Toyota Inova E 2006	Rear Bumper	22	5	3	4
2	Toyota Inova E 2006	Panel	8	5	4	4
3	Toyota Inova E 2006	Ban	19	6	5	5

Gambar 4.14 Implementasi Hasil Perhitungan *Min Max 4*

Permintaan Rata2	Lead Time (Hari)	Safety Stock	Min	Max	Q	Aksi
3	4	8	20	24	4	Restok Spare Part
4	4	4	20	32	12	Restok Spare Part
5	5	5	30	50	20	Restok Spare Part

Gambar 4.15 Implementasi Hasil Perhitungan *Min Max* 5

Pada Gambar 4.15 jika ditekan tombol berwarna kuning “Restok Spare Part” maka akan diarahkan ke halaman pemesanan *spare part* dengan otomatis *spare part* dan jumlah (dari nilai Q) yang harus dipesan terisi, sesuai dengan baris data yang dipilih, untuk implementasi dapat dilihat pada Gambar 4.16.

Gambar 4.16 Pemesanan Melalui Pengendalian Persediaan

#### 4.4. Hasil *Testing*

*Testing* utama berfokus mengenai pengendalian persediaan dengan hasil *testing* atau implementasi seperti pada Gambar 4.10, sampai Gambar 4.16. Berikut merupakan hasil *testing* yang sudah dilakukan (*Black Box Testing* dan *User Acceptance Testing*).

#### 4.4.1. Black Box Testing

Berikut ini adalah hasil pengujian fungsi aplikasi dengan menggunakan *blacbox testing* sesuai dengan fungsi yang ada pada skenario, dapat dilihat pada Tabel 4.6. Untuk lebih detailnya dapat dilihat pada Lampiran 8.

Tabel 4.6 *Black Box Testing*

No	Fungsi	Presentase Keberhasilan
1	<i>Login</i>	100%
2	<i>Permintaan Spare Part</i>	100%
3	<i>Melihat Data Spare Part</i>	100%
4	<i>Pengelolaan Data Master</i>	100%
5	<i>Spare Part Keluar</i>	100%
6	<i>Pemesanan Spare Part</i>	100%
7	<i>Penerimaan Spare Part</i>	100%
8	<i>Retur Spare Part</i>	100%
9	<i>Melihat Log Spare Part</i>	100%
10	<i>Laporan</i>	100%
11	<i>Melihat Hasil Perhitungan Min Max</i>	100%

Berikut ini adalah perhitungan manual yang dilakukan dengan menggunakan data yang ada pada implementasi Gambar 4.12 dan Gambar 4.13. Didapatkan hasil perhitungan manual dan aplikasi memiliki hasil sama yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Contoh Perhitungan Manual Implementasi

Nama Spare Part	Safety Stock	Min	Max	Q
<i>Pad, Rear Bumper F</i>	$(4 - 3) \times 5 = 5$	$(3 \times 5) + 5 = 20$	$2 \times (3 \times 5) = 30$	$30 - 20 = 10$
<i>Rear Bumper</i>	$(5 - 3) \times 4 = 8$	$(3 \times 4) + 8 = 20$	$2 \times (3 \times 4) = 24$	$24 - 20 = 4$
<i>Panel</i>	$(5 - 4) \times 4 = 4$	$(4 \times 4) + 4 = 20$	$2 \times (4 \times 4) = 32$	$32 - 20 = 12$
<i>Ban</i>	$(6 - 5) \times 5 = 5$	$(5 \times 5) + 5 = 30$	$2 \times (5 \times 5) = 50$	$50 - 30 = 20$
<i>Lampu Headlamp</i>	$(5 - 4) \times 4 = 4$	$(4 \times 4) + 4 = 20$	$2 \times (4 \times 4) = 32$	$32 - 20 = 12$

#### 4.4.2. User Acceptance Testing (UAT)

UAT dilakukan kepada 2 *Service Advisor* dan 3 Administrasi pada bengkel Modifie, dengan *user* menjalankan aplikasi sesuai skenario, kemudian mengisi kuisioner yang diberikan, berikut merupakan hasil dari *User Acceptance Testing* (UAT) dari 2 *Service Advisor* (SA) dan 3 Administrasi dapat dilihat pada Tabel 4.8 dan Tabel 4.9.

Tabel 4.8 *User Acceptance Testing (Service Advisor)*

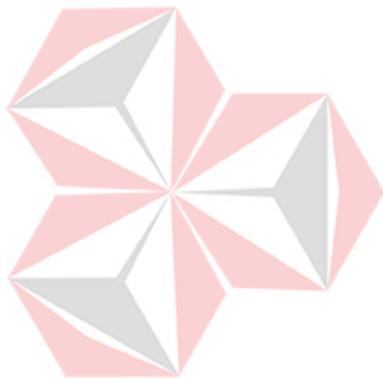
Nama	Persentase Penerimaan Aplikasi
Erik Setiawan	100%
Jeffry	100%

Berikut merupakan hasil dari *User Acceptance Testing* (UAT) dan 3 Administrasi pada Tabel 4.9. Untuk lebih detailnya dapat dilihat pada Lampiran 9.

Tabel 4.9 *User Acceptance Testing* (Administrasi)

<b>Nama</b>	<b>Persentase Penerimaan Aplikasi</b>
Wiwik Rachmawati	100%
Katherine Putri Eka Permatasari	100%
Lia	100%

Namun didapatkan 3 *user* memberikan masukan pada aplikasi seharusnya ditambahkan fitur untuk membuat SPK (Surat Perintah Kerja) otomatis disaat dilakukan proses permintaan, dan juga ditambahkan pelanggan (daftar pelanggan) pada permintaan.



UNIVERSITAS  
**Dinamika**

## BAB V PENUTUP

### 5.1. Kesimpulan

Berdasarkan pada uji coba pada aplikasi pengendalian persediaan bengkel Modifie menggunakan metode *min max* dapat ditarik kesimpulan yaitu:

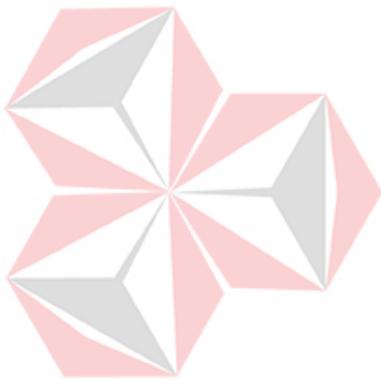
1. Aplikasi dapat membantu dalam pengendalian persediaan menggunakan metode *min max* dengan memberikan notifikasi, daftar, dan jumlah *spare part* yang stoknya mendekati, atau kurang dari nilai *minimum stock* pada halaman *dashboard*.
2. Aplikasi dapat memberikan peringatan stok *spare part* mendekati atau kurang dari nilai *minimum stock* ketika pengguna akan melakukan konfirmasi *spare part* keluar dan juga akan menonaktifkan tombol konfirmasi apabila stok sama atau kurang dari nilai *safety stock*, serta memberikan informasi apakah sudah dilakukan pemesanan atau tidak.
3. Aplikasi bisa memberikan jumlah yang harus dipesan agar mencapai stok optimal.
4. Aplikasi dapat melakukan proses rekap data *spare part* yang membantu dalam pengarsipan pemesanan dan pengecekan ketersediaan *spare part*.
5. Aplikasi dapat membuat laporan (pdf) pada transaksi *spare part* dari *spare part* keluar, pemesanan, penerimaan, dan retur *spare part*.
6. Efisiensi dari *Service Advisor* (SA) yang memberikan *list spare part* yang dibutuhkan ke Administrasi dapat dipersingkat dengan hanya melakukan *input* data permintaan.

### 5.2. Saran

Berdasarkan dari penelitian yang telah dilakukan, penulis memberikan saran untuk pengembangan penelitian kedepan yaitu:

1. Aplikasi dapat ditambahkan fitur pembuatan surat perintah kerja secara otomatis di saat dilakukan proses permintaan, dan pada permintaan ditambahkan daftar pelanggan.

2. Aplikasi dapat ditambahkan fitur pemesanan secara langsung ke *supplier*.
3. Aplikasi bisa ditambahkan metode *Decision Support System* (DSS) untuk memilih *supplier* terbaik dengan membandingkan harga, dan *lead time* dari *supplier* atau bisa ditambahkan penerapan metode peramalan untuk mendapatkan hasil yang lebih akurat.
4. Pada bengkel Modifie sebaiknya ditambahkan bagian gudang, dan *purchasing* untuk mengelola bagian persediaan, dan pemesanan *spare part* agar Administrasi bisa berfokus pada bagian administratif bengkel.

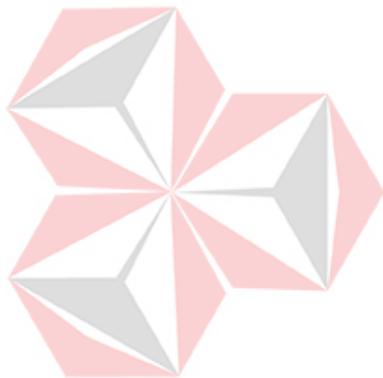


UNIVERSITAS  
Dinamika

## DAFTAR PUSTAKA

- Amrin, Larasati, M., & Satriadi, I. (2020). Model Waterfall Untuk Pengembangan Sistem Informasi Pengolahan Nilai Pada Smp Kartika Xi-3 Jakarta Timur. *Jurnal Teknik Komputer AMIK BSI*, 135- 140.
- Aulia, N., & Indrayanti, A. (2021). Pengujian Aplikasi Mahasiswa Stmik Jakarta Sti&k Menggunakan Metode Blackbox Testing. *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K*, 229-239.
- Bolung, M., & Tampangela, H. (2017). Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak. *Jurnal ELTIKOM, Vol. 1 No. 1*, 1 - 10.
- Hady, E., Haryono, K., & Rahayu, N. (2019). User Acceptance Testing (UAT) Pada Purwarupa Sistem Tabungan Santri (Studi Kasus: Pondok Pesantren Al-Mawaddah). *Jurnal Ilmiah Multimedia dan Komunikasi, Volume 5 Nomor 1*, 1-10.
- Ismawati, K. (2019). Classic Problems: Pengendalian Persediaan. *Ekonomi Bisnis Dan Kewirausahaan Vol VIII*, 12 - 20.
- Leidiyana, H., & Anugrah, A. (2020). Aplikasi Pengendalian Persediaan Barang Berbasis Android Dengan Metode Economic Order Quantity (EOQ) Pada Bengkel Dunia Motor. *Jurnal Komtika (Komputasi dan Informatika)*, 51 - 58.
- Naim, M. A., & Donoriyanto, D. (2020). Pengendalian Persediaan Obat Di Apotek XYZ Dengan Menggunakan Simulasi Monte Carlo. *Jumiten (Jurnal Manajemen Industri dan Teknologi)*, 01-11.
- Nisa, A. (2019). Analisis Pengendalian Persediaan Obat Berdasarkan Metode ABC, EOQ Dan ROP (Studi Kasus Pada Gudang Farmasi Rumah Sakit Muhammadiyah Gresik). *Jurnal Manajerial*, 17-24.
- Permana, R., & Sahara, S. (2018). Penerapan SDLC Waterfall Berbasis Web Pada Toko Giant Komputer Depok. *Jurnal Sistem Informasi Stmik Anak Bangsa Vol.VII No 2*, 205-210.
- Prabawa, G., Darmawiguna, I., & Wirawan, I. (2018). Pengembangan Sistem Pendukung Keputusan Pengendalian Persediaan Barang Menggunakan Metode Economic Order Quantity (EOQ) Dan Min-max Berbasis Web (Studi Kasus : Apotek Sahabat Kita). *Jurnal Nasional Pendidikan Teknik Informatika*, 107 - 120.
- Ryando, D., & Susanti, W. (2019). Penerapan Metode Economic Order Quantity (EOQ) Untuk Menentukan Safety Stock Dan Reorder Point (Studi Kasus Pt. Sinar Glassindo Jaya). *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi Vol. 1 No. 1*, 76- 84.

- Siboro, F., & Nasution, R. (2020). Analisis Pengendalian Persediaan Bahan Baku Dengan Menggunakan Metode Economic Order Quantity (EOQ) Dan Metode Min-Max. *JITEKH, Vol. 8, No. 1,* 34-40.
- Sofi'i, M., Setianti, N., & Purbasari, W. (2020). Sistem Informasi Akuntansi Pengendalian Persediaan Pada Sejati Bengkel Purwokerto. *Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen*, 30-35.
- Wahyudin, & Kristiyanto, F. (2019). Rancang Bangun Sistem Informasi Persediaan Spare Part Mobil Berbasis Web Menggunakan Metode ABC. *Jurnal Infortech*, 9 - 13.
- Widiyanto, A. (2021). Analisis Pengendalian Persediaan Pakan Udang Dengan Metode Min-Max Stock Pada Cv. Ikhsan Jaya. *Jurnal PENA Vol.35 No.1*, 1-10.



UNIVERSITAS  
**Dinamika**