



RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN *SPARE PART* PADA BENGKEL PILANG RAYA MENGGUNAKAN METODE *MIN MAX*



UNIVERSITAS
Dinamika

Oleh :

HAFIZD BIMA ALMUHARRAM

19.41010.0108

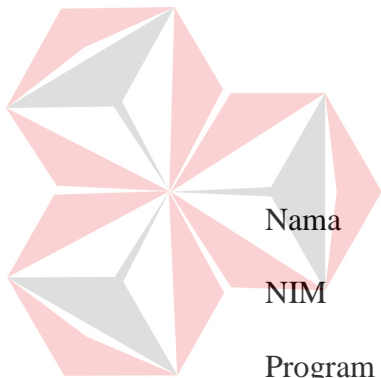
FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN
SPARE PART PADA BENGKEL PILANG RAYA MENGGUNAKAN
METODE MIN MAX**

Diajukan sebagian salah satu syarat untuk menyelesaikan
Program Sarjana



Nama

Oleh :
UNIVERSITAS
Dinamika
: Hafizd Bima Almuhammad

NIM

: 19410100108

Program

: S1 (Strata Satu)

Jurusan

: Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2023

Tugas Akhir

RANCANG BANGUN APLIKASI PENGENDALIAN PERSEDIAAN SPARE PART PADA BENGKEL PILANG RAYA MENGGUNAKAN METODE *MIN MAX*

Dipersiapkan dan disusun oleh

Hafizd Bima Almuhammad

NIM: 19.41010.0108

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada: Selasa, 02 Februari 2023

Susunan Dewan Pembahas

Pembimbing

I. Sulistiowati, S.Si., M.M.
NIDN. 0719016801

II. Ir. Henry Bambang Setyawan, M.M.
NIDK. 8973650022

Pembahas

I. Dr. Drs. Antok Supriyanto, M.MT.
NIDN. 0726106201



Digitally signed
by Henry
Bambang S

Digitally signed by Antok
Supriyanto
DN: cn=Antok Supriyanto,
o=Universitas Dinamika, ou=FEB,
email=antok@dinamika.ac.id, c=ID
Date: 2023.02.02 21:54:38 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana



Digitally signed by
Universitas Dinamika
Date: 2023.02.03
10:43:04 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA



UNIVERSITAS
Be your self and confident.
-Hafizd Bima Almuhammad-

Dinamika



UNIVERSITAS
Laporan Tugas Akhir ini
Saya persembahkan kepada
Keluarga, Dosen Pembimbing, dan
Teman-teman tercintah saya.

Dinamika

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Hafizd Bima Almuhammad

NIM : 19410100108

Program Studi : S1 Sistem Informasi

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya : **RANCANG BANGUN APLIKASI PENGENDALIAN
PERSEDIAAN SPARE PART PADA BENGKEL
PILANG RAYA 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, 03 Januari 2023

Yang menyatakan



Hafizd Bima Almuhammad

NIM: 19410100108

ABSTRAK

Bengkel Pilang Raya adalah perusahaan yang menyediakan jasa perbaikan, penjualan *spare part* dan aksesoris mobil. Salah satu jasa perbaikan yang paling banyak dicari adalah layanan perbaikan AC mobil, karena bengkel tersebut merupakan spesialis dalam hal tersebut. Data bulan Januari hingga Agustus 2022 menunjukkan bahwa rata-rata jumlah mobil yang datang untuk perbaikan sebanyak 87 mobil, sementara jumlah mobil yang sudah diperbaiki hanya 76 mobil. Hal ini menyebabkan kekecewaan pelanggan yang memutuskan pindah ke bengkel lain. Untuk mengatasi masalah tersebut, maka solusi yang diberikan adalah membuat rancang bangun aplikasi pengendalian persediaan *spare part* dengan menggunakan metode *min max*. Metode ini memiliki keunggulan yang digunakan untuk mengelola *spare part* seperti menentukan persediaan maksimum, minimum, *safety stock*, dan jumlah pemesanan yang dapat mengoptimalkan persediaan. Hasil penelitian berupa aplikasi yang memiliki fitur untuk mengendalikan persediaan *spare part*. Fitur-fitur dalam aplikasi ini meliputi: 1) transaksi *spare part*, 2) pembuatan laporan, 3) perhitungan metode *min max*, 4) pengelolaan *data master* yang berisi data *spare part*, *supplier* dan mobil. Aplikasi ini telah diuji dengan *black box testing*, yang hasilnya 100% semua fungsi sudah berjalan dengan baik. Aplikasi juga diuji dengan UAT, yang hasilnya sangat setuju yang artinya aplikasi sesuai dengan kebutuhan pengguna.

Kata Kunci: bengkel, metode *min max*, persediaan

KATA PENGANTAR

Puji syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul "Rancang Bangun Aplikasi Pengendalian Persediaan Sparepart Pada Bengkel Pilang Raya Menggunakan Metode Min Max". Penyelesaian Tugas Akhir ini juga merupakan hasil dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih dan rasa hormat kepada:

1. Ayah, Ibu, dan seluruh keluarga besar yang selalu mendukung dan memberikan semangat.
2. Bapak Achmad Bintoro, selaku Kepala Bengkel Pilang Raya yang memfasilitasi tempat penelitian Tugas Akhir.
3. Ibu Tri Sagirani, S.Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika yang memberikan informasi terkait pengerjaan Tugas Akhir.
4. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng. selaku Ketua Program Studi S1 Sistem Informasi yang memberikan pengesahan Tugas Akhir.
5. Ibu Sulistiowati, S.Si., M.M. dan Bapak Ir. Henry Bambang Setyawan, M.M. selaku Dosen Pembimbing Tugas Akhir yang membimbing dan memberikan masukan.
6. Bapak Dr. Drs. Antok Supriyanto, M.MT. selaku Dosen Penguji Tugas Akhir yang menguji dan memberikan masukan.
7. Teman-teman yang memberikan bantuan dan dukungan dalam penyusunan laporan.

Penulis menyadari bahwa Tugas Akhir masih jauh dari sempurna dan dengan demikian, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan laporan di masa yang akan datang. Penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi penulis dan para pembaca.

Surabaya, 02 Februari 2023

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	4
1.4. Tujuan	4
1.5. Manfaat	4
BAB II LANDASAN TEORI	5
2.1. Penelitian Terdahulu	5
2.2. Bengkel	6
2.3. Website	6
2.3. Pengendalian Persediaan	7
2.4. Metode <i>Min Max</i>	7
2.5. <i>Black Box Testing</i>	8
2.6. <i>System Development Life Cycle (SDLC)</i>	8
2.7. <i>User Acceptance Testing (UAT)</i>	9
BAB III METODOLOGI PENELITIAN	11
3.1. <i>Communication</i>	11
3.1.1. <i>Project Initiation</i>	11
3.1.2. <i>Requirement Gathering</i>	13
3.2. <i>Planning</i>	25
3.3. <i>Modeling</i>	25
3.3.1. <i>System flow</i>	25
3.3.2. <i>Data Flow Diagram (DFD)</i>	26

3.3.3. Diagram Berjenjang	26
3.3.4. <i>Data flow diagram level 0</i>	26
3.3.5. <i>Data flow diagram level 1</i>	28
3.3.6. <i>Data flow diagram level 2</i>	28
3.3.7 <i>Conceptual data model</i>	30
3.3.8 <i>Physical data model</i>	30
3.3.9 Desain <i>Interface</i>	31
3.3.9 Desain <i>Input</i>	31
3.3.10 Desain <i>Output</i>	31
3.4. <i>Construction</i>	31
3.4.1. <i>Coding</i>	32
3.4.2. <i>Testing</i>	32
BAB IV HASIL DAN PEMBAHASAN	35
4.1 Implementasi	35
4.2 Implementasi Sistem	35
4.3. Hasil <i>Testing</i>	37
4.3.1. <i>Black Box Testing</i>	37
4.3.2. <i>User Acceptance Testing (UAT)</i>	37
BAB V PENUTUP	39
5.1. Kesimpulan.....	39
5.2. Saran.....	39
DAFTAR PUSTAKA	40
LAMPIRAN	42

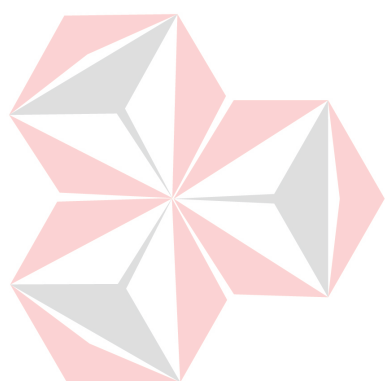
DAFTAR TABEL

Tabel 1.1 Data perbaikan mobil	2
Tabel 1.2 Data <i>spare part</i>	2
Tabel 2.1 Penelitian Terdahulu	5
Tabel 3.1 Identifikasi masalah	14
Tabel 3.2 Identifikasi pengguna	14
Tabel 3.3 Analisis Kebutuhan Pengguna	16
Tabel 3.4 Analisis Kebutuhan Fungsional	17
Tabel 3.5 Analisis Kebutuhan Sistem	18
Tabel 3.6 Simulasi <i>Min Max</i>	22
Tabel 3.7 Skenario <i>Black Box Testing</i> Login	32
Tabel 3.8 Skenario <i>Black Box Testing</i> Administrasi Persediaan	32
Tabel 3.9 Skenario UAT Administrasi Persediaan	33
Tabel 4.1 Hasil <i>Black Box Testing</i>	37
Tabel 4.2 Kriteria UAT	37
Tabel 4.3 Hasil Responden UAT Administrasi Persediaan	38
Tabel 4.4 Hasil Responden UAT Pemilik Bengkel	38
Tabel L1.1 Jadwal Kerja	42
Tabel L8.1 Hasil <i>Black Box Testing</i>	57
Tabel L9.1 Hasil UAT	58

DAFTAR GAMBAR

Gambar 2.1 Fase <i>Waterfall</i>	8
Gambar 3.1 Tahapan Penelitian	11
Gambar 3.2 Proses bisnis dengan flow diagram.	15
Gambar 3.3 Diagram IPO (1).....	19
Gambar 3.4 Diagram IPO (2).....	20
Gambar 3.5 Context diagram	26
Gambar 3.6 System Flow Diagram.....	27
Gambar 3.7 data flow diagram level 0	28
Gambar 3.8 Diagram berjenjang	29
Gambar 3.9 Conceptual data model	30
Gambar 3.10 Physical data model.....	31
Gambar 4.1 Implementasi Hasil Perhitungan Min Max	36
Gambar L3.1 Dfd level 1 pengelolaan data master.....	45
Gambar L3.2 Dfd level 2 <i>data master</i> mobil	45
Gambar L3.3 Dfd level 2 <i>data master</i> spare part.....	46
Gambar L3.4 Dfd level 2 <i>data master</i> supplier.....	46
Gambar L3.5 Dfd level 1 transaksi <i>spare part</i>	47
Gambar L3.6 Dfd level 1 perhitungan <i>min max</i>	47
Gambar L3. 7 Dfd level 1 laporan	48
Gambar L4.1 Interface Login.....	49
Gambar L4.2 Desain <i>Interface Supplier</i>	49
Gambar L4.3 Desain <i>Interface</i> Daftar Penerimaan.....	50
Gambar L4.4 Desain <i>Interface</i> Tambah Permintaan.....	50
Gambar L4.5 Desain <i>Interface</i> beranda	51
Gambar L5.1 Desain <i>input</i> faktur pembelian <i>spare part</i>	52
Gambar L5.2 Desain <i>input</i> faktur perbaikan mobil	52
Gambar L5.3 Desain <i>input</i> faktur permintaan <i>spare part</i>	53
Gambar L5.4 Desain <i>input</i> surat pengantar pembelian <i>spare part</i>	53
Gambar L6.1 Desain <i>output</i> faktur pemesanan <i>spare part</i>	54
Gambar L6.2 Desain <i>output</i> daftar spare part yang dipesan	54

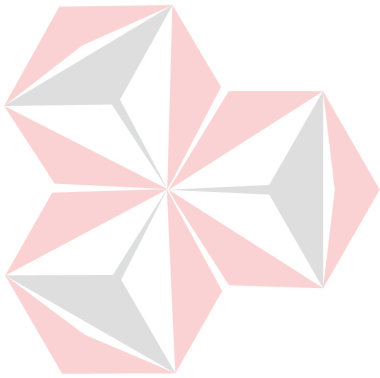
Gambar L7.1 Implementasi <i>login</i>	55
Gambar L7.2 Implementasi Melihat Daftar <i>Spare Part</i>	55
Gambar L7.3 Implementasi Data Permintaan	55
Gambar L7.4 Implementasi Tambah Pemesanan.....	56
Gambar L7.5 Implementasi Konfirmasi Penerimaan.....	56
Gambar L7.6 Implementasi Laporan	56



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1. Jadwal Kerja	42
Lampiran 2. Hasil Wawancara	43
Lampiran 3. DFD Level 1 dan DFD Level 2	45
Lampiran 4. Desain <i>Interface</i>	49
Lampiran 5. Desain <i>Input</i>	52
Lampiran 6 Desain <i>Output</i>	54
Lampiran 7 Implementasi Sistem	55
Lampiran 8 Hasil <i>Black Box Testing</i>	57
Lampiran 9 Hasil <i>User Accepted Testing</i>	58
Lampiran 10 Cek Plagiasi	60
Lampiran 11 Biodata	61



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan informasi dan teknologi di era digital yang semakin pesat telah membawa berbagai perubahan, khususnya untuk membantu kegiatan manusia. Salah satu dampak perkembangan teknologi terdapat pada pengendalian persediaan barang, seperti pengelolaan data persediaan, termasuk barang masuk dan keluar dari pelanggan dan *supplier*, berapa persediaan yang harus diisi kembali dan memberikan informasi yang berupa laporan mengenai transaksi pada *spare part*.

Bengkel Pilang Raya adalah bengkel pribadi. Ahmad Bintoro merupakan *owner* Bengkel Pilang Raya. Bengkel Pilang Raya dapat memberikan layanan terbaik bagi para pelanggannya. Bengkel ini menawarkan berbagai jasa perbaikan, salah satunya adalah perbaikan AC mobil, karena Bengkel Pilang Raya merupakan spesialis dalam bidang tersebut. Selain jasa perbaikan, Bengkel Pilang Raya juga menjual berbagai macam *spare part* dan aksesoris mobil, yang cocok untuk berbagai merk kendaraan.

Bulan Januari sampai Agustus 2022 menunjukkan data rata-rata mobil yang datang untuk melakukan perbaikan sekitar 87 mobil, dan rata-rata jumlah mobil yang sudah diperbaiki sekitar 76 mobil. Untuk lebih rinci data bisa dilihat pada Tabel 1.1.

Proses bisnis yang dilakukan pada Bengkel Pilang Raya dengan baik yang dimulai dengan menerima keluhan perbaikan mobil kepada administrasi, kemudian melakukan pengecekan mobil melalui teknisi dan perbaikan mobil, melakukan pembayaran kepada administrasi. Selain itu, administrasi juga mengelola pencatatan *spare part*. Pencatatan *spare part* dilakukan untuk proses keluar masuk *spare part* yang akan digunakan untuk perbaikan mobil.

Tabel 1.1 Data perbaikan mobil

Periode 2022	Jumlah Mobil datang untuk melakukan perbaikan/unit	Jumlah mobil yang sudah diperbaiki/unit	Jumlah mobil yang tidak dilayani karna kehabisan spare part/unit
Januari	50	39	11
Februari	40	34	6
Maret	61	50	11
April	132	131	1
Mei	129	123	6
Juni	90	80	10
Juli	80	61	19
Agustus	112	102	10
Rata - rata	87	78	9

Proses masuknya *spare part* yaitu proses pemesanan *spare part* dari *supplier* dengan mendatangi ke toko *spare part* dan membeli sesuai kebutuhan bengkel. Proses keluarnya *spare part* dilakukan ketika teknisi menggunakan *spare part* untuk memenuhi permintaan kebutuhan mobil pelanggannya.

Permasalahannya adalah dari data mobil datang untuk melakukan perbaikan dan mobil yang sudah diperbaiki didapatkan sekitar 9 rata rata mobil pelanggan tidak mendapatkan *spare part* yang sesuai, seperti evaporator, kompresor dan kondensor dikarenakan persediaan tersebut kosong. Hal ini jika dibiarkan maka mengakibatkan pelanggan kecewa, AC mobil tidak mendapatkan perbaikan sehingga berpindah ke bengkel mobil yang lain.

Data bulan januari, terlihat *spare part* evaporator, kondensor dan kompresor kekurangan persediaan pada gudang. Hal ini dapat dilihat dengan jumlah permintaan *spare part* yang lebih besar dibandingkan dengan persediaan yang ada pada gudang. Untuk lebih rinci, data *spare part* dilihat pada Tabel 1.2.

Tabel 1.2 Data *spare part*

Spare part	Periode 2022	Stok spare part/unit	Jumlah permintaan spare part/unit
Evaporator	Januari	50	93
Kondensor	Januari	40	50
Kompresor	Januari	23	41

Permasalahan lainnya adalah, pengecekan persediaan stok *spare part* pada Bengkel Pilang Raya dihitung secara manual, dengan melihat persediaan *spare part*

di gudang bengkel. Hal ini membutuhkan ketelitian dengan mengecek satu persatu agar tidak terjadi kekosongan persediaan *spare part*.

Untuk mengatasi permasalahan yang ada pada bengkel Pilang Raya, solusi yang ditawarkan adalah dengan membuat sebuah aplikasi untuk mengendalikan persediaan *spare part* yang menggunakan metode *min max*. Ada beberapa metode yang dapat digunakan dalam pengelolaan persediaan, seperti metode EOQ (*Economic Order Quantity*) dan *Reorder Point*. Metode EOQ membantu dalam menentukan jumlah pembelian *spare part* pada setiap kali pesan dengan biaya minimum, sementara metode *Reorder Point* membantu menentukan titik dimana jumlah persediaan *spare part* harus dipesan kembali.

Oleh karena itu, dalam penelitian ini metode *min max* dipilih sebagai solusi pengendalian persediaan *spare part*. Metode ini berfokus pada menjaga agar tingkat persediaan barang selalu ada dalam kisaran minimum dan maksimum yang ditentukan, sehingga dapat meminimalisir kerugian dan memaksimalkan keuntungan (Hertanto, 2020). Metode *min max* memiliki acuan jumlah persediaan yang digunakan untuk pengelolaan barang. Acuan jumlah tersebut berupa persediaan maksimum, persediaan minimum, *safety stock* dan jumlah pemesanan yang dapat mengoptimalkan persediaan. Apabila persediaan telah mencapai *safety stock*, maka permintaan *spare part* tidak bisa dilayani, jika dilayani maka akan terjadinya *stock out*. Apabila persediaan jumlahnya mencapai batas persediaan minimum maka dilakukan pemesanan *spare part*. Persediaan maksimum merupakan jumlah maksimum yang disarankan untuk disimpan dalam persediaan. Jumlah pemesanan digunakan sebagai acuan dalam pembelian barang sesuai dengan kebutuhan. Ada empat rumus yang digunakan dalam metode *min max* dalam pengendalian persediaan. Pertama, menentukan jumlah *safety stock*. Kedua, menentukan jumlah persediaan minimum. Ketiga, menentukan jumlah persediaan maksimum. Keempat, menentukan jumlah pemesanan *spare part* untuk memenuhi kebutuhan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang ada, rumusan masalah sebagai berikut: bagaimana rancangan pengendalian persediaan *spare part* pada bengkel Pilang Raya menggunakan metode *min max*.

1.3. Batasan Masalah

Berdasarkan uraian latar belakang, terdapat beberapa batasan masalah, antara lain:

1. Dalam penelitian ini, metode *min max* digunakan untuk menentukan *safety stock*, memperkirakan jumlah persediaan minimal dan maksimal, serta menghitung jumlah pemesanan yang diperlukan untuk periode berikutnya.
2. Apabila *lead time* suatu *spare part* tidak tetap, maka digunakan *lead time* rata rata
3. Aplikasi mengelola persedian *spare part* khusus perbaikan AC, seperti kompresor, kondensor dan evaporator.
4. Aplikasi tidak melibatkan masalah keuangan.

1.4. Tujuan

Berdasarkan latar belakang dan rumusan masalah, tujuan dari penelitian ini adalah menghasilkan aplikasi pengendalian persediaan *spare part* pada Bengkel Pilang Raya yang menggunakan metode *min max*. Tujuannya adalah untuk mencegah kekurangan persediaan *spare part* dan menentukan jumlah pemesanan yang tepat untuk pembelian persediaan.

1.5. Manfaat

Adapun manfaat yang diharapkan dari aplikasi pengendalian persediaan *spare part* pada Bengkel Pilang Raya meliputi:

1. Membantu pengendalian persediaan *spare part*.
2. Memudahkan merekap data persediaan *spare part*.
3. Menyediakan solusi untuk mempermudah pembuatan laporan pengendalian persediaan *spare part*.

BAB II LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian terdahulu yang digunakan sebagai acuan untuk pelaksanaan dan pengembangan penelitian ini memiliki permasalahan yang sama. Namun, solusi dan pemecahan masalahnya berbeda. Beberapa penelitian terdahulu yang relevan dapat ditemukan pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Nama Penelitian	Judul Penelitian	Hasil Penelitian	Perbedaan
(Leidiyana & Anugrah, 2020)	Aplikasi Pengendalian Barang Dengan Metode Economic Order Quantity (EOQ) Pada Bengkel Dunia Motor	Pada jurnal ini dikatakan bahwa implementasi aplikasi yang dibangun dapat digunakan mengetahui besarnya jumlah pembelian persediaan, dan pemesanan yang optimal agar dapat melakukan penghematan untuk pembelian barang.	Pada rancang bangun aplikasi ini tidak menggunakan metode EOQ. Metode EOQ digunakan agar Menstabilkan stok barang dan mengurangi biaya pemesanan dan pemeliharaan barang sampai tingkat minimum yang memungkinkan
(Sayang, Ega Syahputra dkk. 2022)	Rancang Bangun Aplikasi Pengendalian Persediaan Bahan Baku Berbasis Website Menggunakan Metode Safety Stock dengan Perhitungan Service Level pada CV Dwi Teknik	Pada jurnal ini dikatakan Aplikasi mampu memproses data barang, melakukan penambahan dan pengurangan persediaan barang melalui perhitungan metode <i>safety stock</i>	Pada rancangan bangun aplikasi ini, hanya diterapkan satu metode, yaitu pendekatan metode <i>safety stock</i> untuk memberikan notifikasi tentang barang yang telah mencapai <i>safety stock</i> dan menggunakan perhitungan tingkat layanan
(Hamzah, 2022)	Rancang Bangun Aplikasi Pengendalian Barang Berbasis Website dengan Metode Reorder Point Pada CV Djaja Aksa Mandiri	Aplikasi ini dapat menunjukkan jumlah stok barang dan nilai titik pesan ulang yang diperhitungkan dengan bantuan laporan perhitungan yang ditampilkan pada beranda dan notifikasi mengenai barang yang perlu dipesan ulang.	Dalam rancangan aplikasi ini, tidak hanya menggunakan pendekatan Reorder Point saja, di mana suatu barang di gudang harus memiliki tambahan persediaan sebelum mencapai kehabisan

2.2. Bengkel

Bengkel merupakan sebuah fasilitas yang menyediakan ruang dan alat untuk memperbaiki suatu item. Sedangkan perbengkelan adalah keterampilan dan pengetahuan tentang bagaimana menggunakan alat dan metode untuk membuat, memodifikasi, atau memperbaiki suatu item menjadi bentuk yang baru dari segi manfaat maupun tampilan. Dalam kehidupan sehari-hari di Indonesia, istilah "bengkel" seringkali dihubungkan dengan jasa perbaikan pada kendaraan bermotor. Tujuan utama dari bengkel adalah sebagai tempat untuk memperbaiki, mengganti, dan memelihara kendaraan bermotor (Putra, 2021).

2.3. Website

Website didefinisikan sebagai nama atau *subdomain* yang ada di *internet* (Harminingtyas, 2014). Ini adalah *platform* yang menyediakan informasi kepada pengguna melalui *web browser*, yang bersifat statis atau dinamis. Terdiri dari serangkaian halaman yang saling terkait dan terhubung melalui tautan antar halaman. Agar *website* mudah diakses dan menyediakan informasi yang diperlukan oleh pengguna, ada tiga elemen penting yang perlu disiapkan:

1. Domain

Domain adalah alamat unik yang digunakan untuk mengidentifikasi *website* melalui *internet*. *Domain* atau *URL* sering disebut sebagai nama *domain* dan berfungsi sebagai alamat untuk menemukan *website* melalui *internet*.

2. Hosting

hosting adalah layanan yang memberikan ruang untuk menyimpan data, file, gambar, video, *email*, statistik, *database*, dan informasi lain yang ditampilkan pada sebuah *website*. Kapasitas penyimpanan yang disediakan bergantung pada ukuran web hosting yang dimiliki atau disewa oleh pemilik *website*.

3. Desain Website

Desain adalah elemen penting dan utama dalam membuat sebuah *website*. Desain menentukan tingkat kualitas dan estetika suatu *website* dan memiliki pengaruh besar terhadap penilaian pengguna akan kebaikan *website* tersebut. Ada beberapa pilihan untuk membuat desain *website*, seperti melakukan sendiri atau *hiring* jasa seorang desainer *website* profesional.

2.3. Pengendalian Persediaan

Pengendalian adalah upaya untuk memastikan bahwa suatu kegiatan berlangsung sesuai dengan rencana. Persediaan adalah barang-barang yang digunakan dalam suatu usaha perusahaan. Pengendalian perusahaan adalah tindakan untuk memastikan bahwa bahan-bahan dan barang-barang yang menjadi objek usaha perusahaan dapat berjalan sesuai dengan rencana. Bahan baku adalah sumber dari alam atau dari pemasok yang akan diubah menjadi produk jadi melalui proses-proses di dalam perusahaan (Wahid & Munir, 2020).

2.4. Metode *Min Max*

Metode *Min Max* adalah salah satu metode pengendalian persediaan yang digunakan untuk memastikan bahwa persediaan tidak kehabisan atau terlalu banyak. Metode ini bekerja dengan memonitor persediaan minimum dan maksimum. Bila persediaan sudah mencapai batas minimum, maka perusahaan harus melakukan pemesanan ulang untuk memastikan bahwa jumlah persediaan selalu dalam kondisi optimal. Batas minimum menunjukkan tingkat *re-order level*, sementara batas maksimum adalah jumlah maksimal yang dapat diterima oleh perusahaan sebagai investasi dalam bentuk persediaan bahan baku. Melalui persediaan yang baik pada batas minimum dan maksimum, perusahaan dapat menentukan jumlah pemesanan yang tepat. (Putri & Ulkhaq, 2017). Rumus *Min Max* adalah sebagai berikut:

$$SS = (\text{Permintaan maksimum} - R) \times L$$

$$\text{Min} = (R \times L) + SS$$

$$\text{Max} = 2(R \times L) + SS$$

$$Q = \text{Max} - \text{Min}$$

Keterangan :

SS : Jumlah persediaan pengaman

Q : Jumlah pemesanan

Min : Persediaan minimum

Max : Persediaan maksimum

R : Permintaan barang rata-rata per periode.

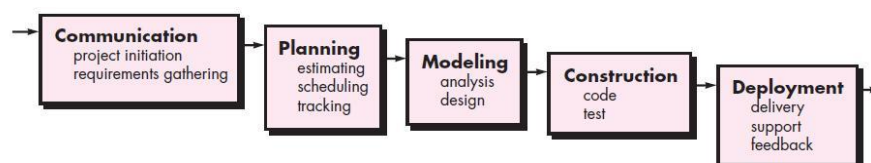
L : Waktu pesanan dalam satuan waktu hitungan hari

2.5. Black Box Testing

Menurut Rahmanila (2014), Metode pengujian *Black Box* adalah suatu pendekatan pengujian yang memfokuskan pada *output* dan fungsionalitas aplikasi tanpa mengacu pada implementasi internal. Metode ini digunakan untuk menemukan masalah atau kesalahan sebelum aplikasi dirilis ke publik. Salah satu jenis pengujian *black box* adalah pengujian fungsional, yang bertujuan untuk menguji fitur atau fungsionalitas spesifik dari aplikasi dengan memastikan bahwa *output* yang dihasilkan sesuai dengan *input* dari pengguna.

2.6. System Development Life Cycle (SDLC)

Proses pengembangan software dikenal dengan *Software Development Life Cycle* (SDLC) adalah metode yang menjelaskan langkah-langkah sistematis dan berurutan dalam membangun aplikasi. Beberapa tahap dalam SDLC meliputi analisis, desain, implementasi, dan perawatan, yang membantu memastikan bahwa software memenuhi harapan pengguna, bekerja secara efektif dan efisien, dan biaya rendah dalam hal perawatan dan pengembangan lanjut. Salah satu model SDLC yang populer dan sering digunakan adalah model *Waterfall*, yang mengikuti urutan sistematis dalam membangun *software*. Model ini juga dikenal sebagai "*Linear Sequential Model*" atau "*Classic Life Cycle*" (Pressman, 2015). Referensi Pressman menjelaskan fase-fase dalam model *Waterfall* pada Gambar 2.1.



Gambar 2.1 Fase *Waterfall*

(Sumber: Pressman, 2015)

a. Communication

Komunikasi yang efisien dengan pemilik bengkel adalah kunci sukses dalam memulai suatu proyek teknis. Langkah awal dalam komunikasi ini adalah menentukan masalah dan mengumpulkan data yang diperlukan, seperti melalui

analisis dan menetapkan fitur dan fungsi *software*. Informasi tambahan dapat ditemukan melalui sumber seperti jurnal ilmiah, artikel, dan sumber daya *internet*.

b. *Planning*

Langkah setelah identifikasi permasalahan adalah proses perencanaan yang fokus pada estimasi tugas teknis, identifikasi risiko potensial, pengidentifikasian sumber daya yang dibutuhkan, definisi produk akhir yang diharapkan, penjadwalan pekerjaan, dan pemantauan progres pembuatan sistem.

c. *Modeling*

Tahap ini bertujuan untuk memahami gambaran besar dari proyek yang akan dikerjakan melalui proses pemodelan arsitektur sistem. Dalam tahap ini, fokus diberikan pada rancangan struktur data, arsitektur *software*, tampilan antarmuka, dan algoritma program untuk memastikan bahwa solusi yang diterapkan sesuai dengan kebutuhan dan standar yang ditentukan.

d. *Construction*

Tahap konstruksi merupakan proses transformasi dari desain menjadi kode yang bisa dipahami oleh mesin. Tahap pengujian dilakukan setelah pengkodean selesai untuk memastikan bahwa sistem berfungsi dengan baik dan untuk menemukan dan memperbaiki kesalahan atau *bug*. Tujuannya adalah untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan spesifikasi yang ditentukan.

e. *Deployment*

Tahap *Deployment* memperkenalkan *software* ke publik dan melibatkan aktivitas untuk memastikan ketersediaan dan kinerja yang optimal. Ini termasuk melakukan pemeliharaan berkala, memperbaiki kesalahan jika diperlukan, mengevaluasi *software*, dan memperbarui *software* berdasarkan umpan balik dari pengguna. Tujuannya adalah memastikan bahwa *software* dapat berfungsi dengan baik dan terus berkembang sesuai dengan peran dan fungsinya.

2.7 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahap pengujian *software* terakhir dan sangat penting. Dalam tahapan ini, sistem diuji untuk menentukan apakah telah memenuhi kebutuhan pengguna dan mampu menjalankan semua skenario bisnis dan pengguna. UAT dilakukan oleh klien dan pengguna akhir. Proses UAT juga

sangat erat kaitannya dengan pembayaran yang diterima oleh pengembang software (Haryono & Rahayu, 2019). Terdapat tiga aspek dalam membuat UAT, seperti:

a. Perencanaan

Perencanaan untuk tahap *User Acceptance Testing* (UAT) harus dimulai sejak awal proyek, karena ada beberapa keputusan dan persiapan yang perlu dilakukan sepanjang proses. Jika perencanaan baru dilakukan di akhir proyek, maka ada kemungkinan masalah akan muncul yang dapat menyebabkan penundaan dan membuat target waktu penyelesaian proyek tergeser. Untuk melakukan perencanaan UAT, dapat dilakukan dengan membuat *test plan* yang memuat beberapa komponen seperti tanggal, kondisi lingkungan, peran dan tanggung jawab dari pelaku, hasil dan proses analisis, serta kriteria masuk dan keluar.

b. Persiapan

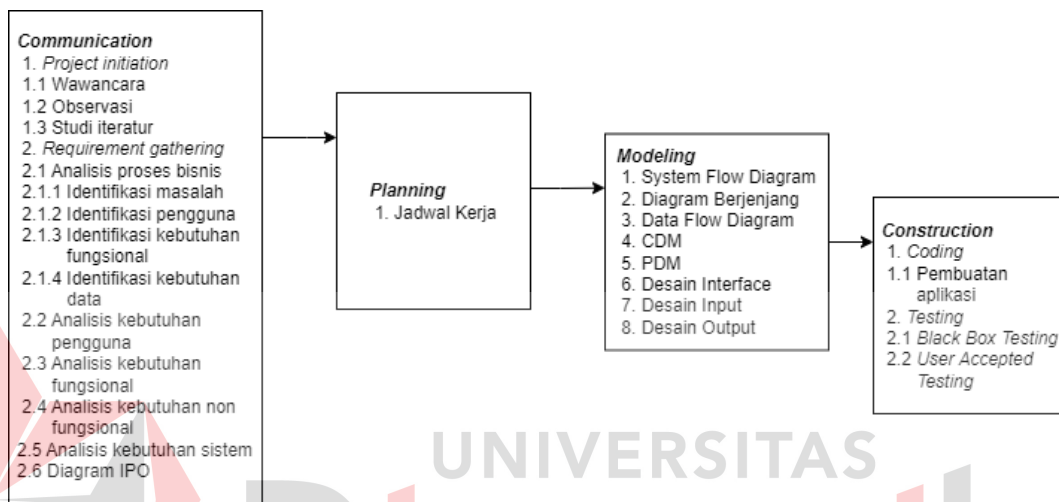
Untuk melakukan pengujian, dibutuhkan data pengujian yang diperoleh melalui sumber daya yang besar dan memerlukan proses pembuatan yang rumit. Data pengujian dapat dibuat dengan cara manual melalui input pengguna atau dengan mengambil data internal dari *database*. Saat data pengujian dimasukkan oleh pengguna, penting untuk memastikan bahwa data tersebut didefinisikan dan didokumentasikan dengan benar, agar sesuai dengan uji yang akan dilakukan. Selain mempersiapkan data pengujian, hal lain yang juga penting untuk dipersiapkan adalah lingkungan bisnis sehari-hari, seperti komputer uji dan lingkungan simulasi bisnis yang dapat melakukan simulasi bisnis yang sesungguhnya.

c. Pengelolaan dan Eksekusi

Dalam melakukan pengujian user acceptance testing (UAT), tanggung jawab utama berada pada pengguna sistem yang bertindak sebagai penguji. Pengguna harus mengidentifikasi tipe kasus yang akan diuji, menyiapkan data pengujian, dan melaksanakan pengujian tersebut. Setelah selesai, pengguna harus membuat laporan hasil pengujian dan menentukan apakah semua persyaratan telah terpenuhi atau ada kesalahan yang harus dibetulkan. Jika terjadi kesalahan, maka perlu dilakukan perbaikan dan dokumentasi untuk memantau perkembangan yang dilakukan.

BAB III METODOLOGI PENELITIAN

Dalam penelitian ini, metodologi yang digunakan adalah Model SDLC *Waterfall*. Metode ini menjadi landasan untuk merancang dan mengembangkan aplikasi atau sistem. Gambar 3.1 menunjukkan langkah-langkah yang akan dilakukan dalam penelitian ini.



Gambar 3.1 Tahapan Penelitian

3.1. Communication

Tahap pertama dalam proses *System Development Life Cycle (SDLC) Waterfall* merupakan tahap komunikasi. Tahap ini dilakukan dengan pemilik Bengkel Pilang Raya untuk memperoleh informasi tentang masalah dalam pengelolaan persediaan bengkel, serta data yang dibutuhkan. Tahap ini terdiri dari dua aktivitas, yaitu *project initiation* dan *requirement gathering*.

3.1.1. Project Initiation

Dalam tahap *Project Initiation* dalam metodologi penelitian ini, terdapat tiga langkah yaitu melakukan Wawancara, Observasi, dan Studi Literatur. Langkah-langkah ini bertujuan untuk mengumpulkan informasi yang diperlukan dan memahami tujuan proyek yang akan dikerjakan.

1) Wawancara

Untuk memahami masalah yang dialami oleh Bengkel Pilang Raya, dilakukan wawancara langsung dengan pemilik bengkel. Sebelum melakukan wawancara, daftar pertanyaan yang berkaitan dengan alur dari proses bisnis, masalah yang ada, dan struktur karyawan bengkel dipersiapkan. Dalam wawancara, pertanyaan yang diajukan berkaitan dengan sistem dan alur proses bisnis di bengkel.

Dari analisis yang sudah dilakukan, maka didapatkan hasil bahwa proses bisnis bengkel memiliki beberapa permasalahan, seperti rata-rata data mobil datang untuk melakukan perbaikan dan mobil yang telah diperbaiki didapatkan sekitar 9 mobil pelanggan tidak mendapatkan *spare part* yang sesuai, dikarenakan persediaan *spare part* kosong. Hal ini jika dibiarkan maka mengakibatkan pelanggan kecewa sehingga berpindah ke bengkel mobil yang lain. Permasalahan lainnya adalah, pengecekan persediaan stok *spare part* pada Bengkel Pilang Raya dihitung secara manual, dengan melihat persediaan *spare part* di gudang bengkel. Hal ini membutuhkan ketelitian dengan mengecek satu persatu agar tidak terjadi kekosongan persediaan *spare part*. Pada lampiran 2, terdapat hasil wawancara.

2) Observasi

Untuk memperoleh informasi beserta data yang berkaitan dengan masalah yang dialami pada Bengkel Pilang Raya, dilakukan observasi terhadap proses bisnis yang ada di bengkel. Observasi dilakukan selama satu bulan dan fokus pada bagian administrasi persediaan. Data yang diperoleh melalui observasi akan digunakan dalam membangun aplikasi dan memahami alur proses bisnis bengkel, serta masalah yang terjadi saat ini.

3) Studi Literatur

Untuk menambah pengetahuan dan memperkuat hasil observasi serta wawancara, peneliti juga melakukan studi literatur dengan membaca jurnal yang relevan. Referensi-referensi tersebut memuat informasi mengenai:

1. Penelitian Terdahulu
2. Bengkel
3. *Website*
3. Pengendalian Persediaan

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

3.1.2. Requirement Gathering

Tahap *Requirement Gathering* digunakan oleh peneliti untuk menganalisis kebutuhan dari sistem yang akan dibuat. Tahap ini didasarkan pada hasil dari wawancara yang dilakukan selama tahap *project initiation*. Dalam tahap ini, peneliti melakukan analisis terhadap proses bisnis, kebutuhan pengguna, kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan sistem. Peneliti juga membuat diagram *Input-Proses-Output (IPO)* sebagai bagian dari tahap ini.

A. Analisis Proses Bisnis

Saat ini Proses bisnis pada bengkel Pilang Raya yang dilakukan secara *offline* dan belum menggunakan teknologi informasi. Pada pengendalian persediaan *spare part*, proses bisnis saat ini dimulai dengan administrasi menerima keluhan perbaikan mobil dari pelanggan. Kemudian admin membuat faktur perbaikan mobil. Lalu teknisi akan melakukan perbaikan sesuai urutan faktur perbaikan mobil. Jika *spare part* tersedia, maka teknisi melakukan perbaikan dan administrasi melakukan transaksi perbaikan dengan pelanggan. Jika *spare part* tidak tersedia, maka administrasi mencatat kebutuhan *spare part* yang akan dibeli. Kemudian pemilik bengkel membeli *spare part* kepada supplier dan melakukan transaksi pembelian *spare part*. Administrasi akan mencatat penerimaan *spare part* yang telah datang dan digunakan teknisi untuk melakukan perbaikan. Proses bisnis saat ini digambarkan dengan flow diagram dapat dilihat pada Gambar 3.2.

1) Identifikasi Masalah

Hasil dari analisis masalah pada proses bisnis dapat ditemukan pada Tabel 3.1 sebagai hasil dari proses analisis yang dilakukan.

Tabel 3.1 Identifikasi masalah

Masalah	Dampak	Solusi
Rata rata permintaan pelanggan yang tidak terpenuhi sekitar 9 mobil karena kehabisan <i>spare part</i> .	Pelanggan berpindah ke bengkel lain.	Merancang dan membangun aplikasi pengendalian persediaan <i>spare part</i> menggunakan metode <i>min max</i> untuk mengoptimalkan persediaan <i>spare part</i> di gudang dan memastikan bahwa persediaan tersebut selalu tersedia sesuai dengan kebutuhan.
Admin memeriksa persediaan <i>spare part</i> secara satu persatu digudang dan manual.	Tidak efisien dan membuang waktu.	Memberikan informasi kepada admin mengenai persediaan <i>spare part</i> yang tersedia.

2) Identifikasi pengguna

Berdasarkan hasil observasi dan wawancara, identifikasi masalah telah dilakukan dan daftar pengguna dari aplikasi dapat ditemukan. Tabel 3.2 menampilkan identifikasi pengguna yang dibangun.

Tabel 3.2 Identifikasi pengguna

No	Pengguna
1.	Administrasi (Bagian persediaan)
2.	Pemilik Bengkel

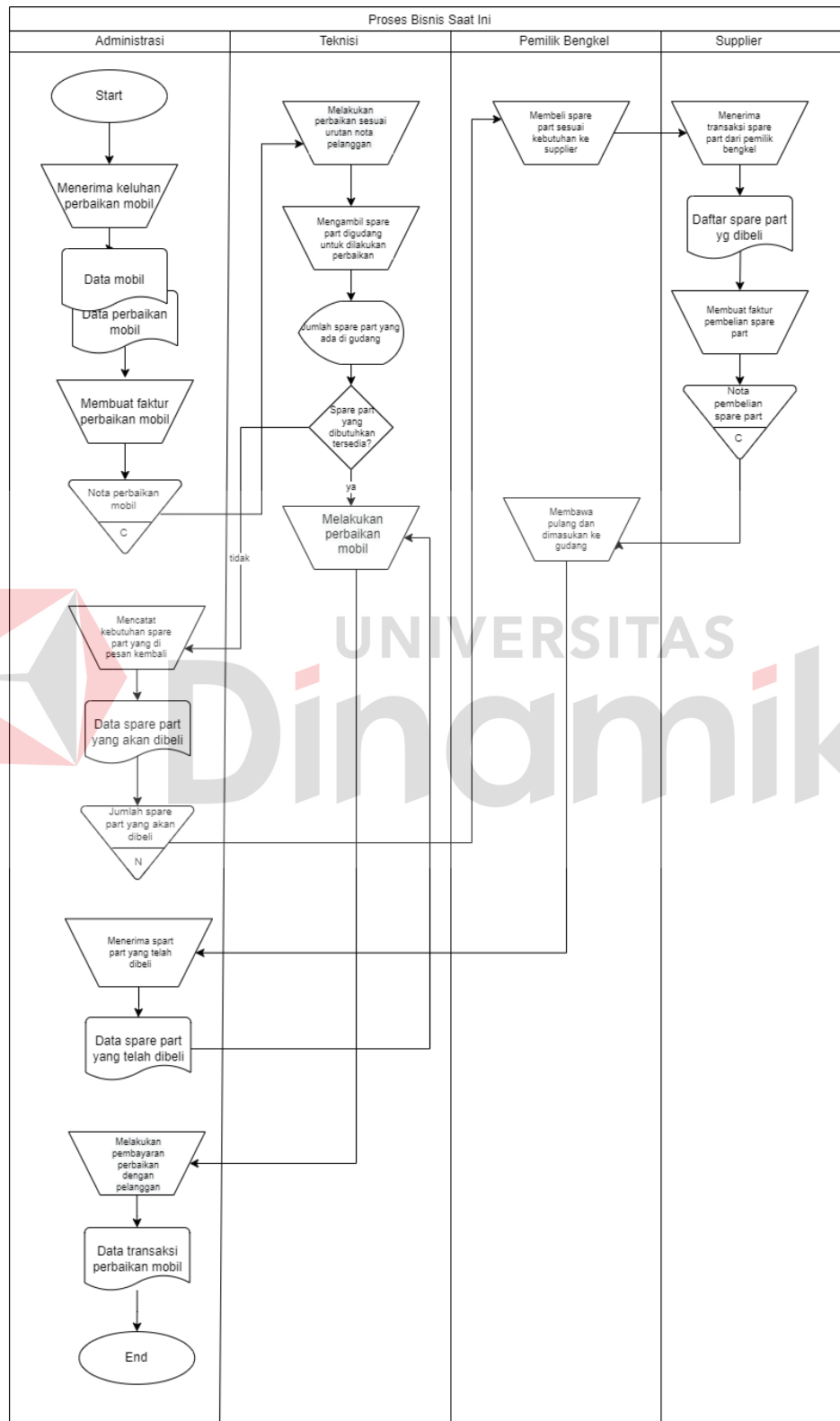
3) Identifikasi kebutuhan fungsional

Berdasarkan hasil observasi, wawancara, dan identifikasi masalah, kebutuhan fungsional dari aplikasi dapat diketahui. Fungsi-fungsi yang diidentifikasi termasuk *login*, pencatatan permintaan *spare part*, pengelolaan data *spare part*, pengelolaan *data master*, pencatatan penerimaan *spare part*, pencatatan pemesanan *spare part*, cetak laporan penerimaan *spare part*, cetak laporan permintaan *spare part*, dan pengendalian persediaan melalui metode *min max*.

4) Identifikasi kebutuhan data

Berdasarkan hasil observasi, wawancara, dan identifikasi masalah, identifikasi data yang dibutuhkan untuk aplikasi dapat dilakukan. Data yang harus dikumpulkan meliputi informasi mengenai *spare part*, mobil, permintaan *spare part*, *supplier*, pemesanan *spare part*, penerimaan *spare part*, laporan penerimaan

spare part, laporan permintaan *spare part*, dan pengelolaan persediaan dengan menggunakan metode *min-max*.



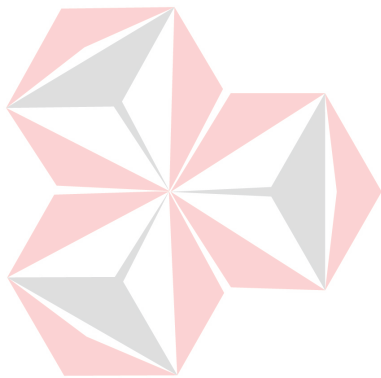
Gambar 3.2 Proses bisnis dengan *flow* diagram.

B. Analisis Kebutuhan Pengguna

Ada dua aktor utama dalam bisnis proses yaitu pemilik bengkel dan administrasi persediaan. Tabel 3.3 menunjukkan kebutuhan pengguna yang ditemukan setelah melalui proses observasi, wawancara, dan identifikasi masalah.

Tabel 3.3 Analisis Kebutuhan Pengguna

Aktor	Kebutuhan Fungsi	Kebutuhan Data	Kebutuhan Informasi
Pemilik Bengkel	<i>Login</i>	<i>Data user</i>	Tahap validasi dilakukan untuk memastikan bahwa informasi <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna sudah cocok dengan catatan sistem.
	Melihat data <i>spare part</i>	<i>Data spare part</i>	- Jumlah <i>spare part</i> - Daftar <i>spare part</i>
Administrasi (bagian persediaan)	<i>Login</i>	<i>Data user</i>	Tahap validasi dilakukan untuk memastikan bahwa informasi <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna sudah cocok dengan catatan sistem.
	Pengelolaan data <i>master</i>	- <i>Data spare part</i> - <i>Data supplier</i> - <i>Data mobil</i>	- Daftar persediaan <i>spare part</i> - Daftar <i>supplier spare part</i> - Daftar mobil pelanggan
	Mencatat penerimaan <i>spare part</i>	- <i>Data supplier</i> - <i>Data spare part</i> - <i>Data pemesanan</i> - <i>Data penerimaan</i>	- Daftar dan jumlah <i>spare part</i> yang diterima - Daftar <i>supplier</i> yang mengirimkan <i>spare part</i> - Jumlah <i>spare part</i> yang dipesan
	Mencatat permintaan <i>spare part</i>	- <i>Data spare part</i> - <i>Data mobil</i> - <i>Data permintaan</i>	- <i>Spare part</i> yang dibutuhkan - Daftar mobil - Permintaan <i>Spare Part</i>
	Mencatat pemesanan <i>spare part</i>	- <i>Data spare part</i> - <i>Data supplier</i> - <i>Data pemesanan</i> - <i>Data detail pemesanan</i>	- Jumlah <i>spare part</i> yang dipesan - Daftar <i>supplier</i> yang mangakodamasi pesanan <i>spare part</i>
	Pengendalian Persediaan <i>spare part</i>	- <i>Data permintaan</i> - <i>lead time spare part</i>	Perhitungan <i>safety stock</i> , persediaan maksimal, persediaan minimum dan jumlah pemesanan
	Membuat laporan	- <i>Data spare part</i> masuk - <i>Data spare part</i> yang digunakan	- Laporan penerimaan <i>spare part</i> - Laporan permintaan <i>spare part</i>



Administrasi
(bagian
persediaan)

UNIVERSITAS
Dianloma

Aktor	Kebutuhan Fungsi	Kebutuhan Data	Kebutuhan Informasi
--------------	-------------------------	-----------------------	----------------------------

C. Analisis Kebutuhan Fungsional

Berikut adalah hasil analisis kebutuhan fungsional yang teridentifikasi dalam Tabel 3.4.

Tabel 3.4 Analisis Kebutuhan Fungsional

Aktor	Fungsi	Deskripsi
Pemilik Bengkel dan Administrasi persediaan	<i>Login</i>	Tahap validasi dilakukan untuk memastikan bahwa informasi <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna sudah cocok dengan sistem.
Administrasi persediaan	Mengelola data <i>master</i>	Fitur <i>data master</i> termasuk opsi untuk membuat, membaca, memperbarui, dan menghapus data.
Administrasi persediaan	Mencatat penerimaan <i>spare part</i>	Memasukkan data <i>spare part</i> yang diterima dari proses pemesanan.
Administrasi persediaan	Mencatat permintaan <i>spare part</i>	Untuk mencatat permintaan <i>sparepart</i> .
Administrasi persediaan	Mencatat pemesanan <i>spare part</i>	Untuk mencatat pemesanan <i>spare part</i>
Administrasi persediaan dan Pemilik Bengkel	Melihat informasi data <i>spare part</i>	Untuk melihat data <i>spare part</i>
Administrasi persediaan	Pengendalian Persediaan dengan metode <i>min max</i>	Untuk Mengendalikan persediaan agar tidak terjadinya kelebihan stok dan kehabisan persediaan dengan menggunakan metode <i>min max</i> .
Administrasi persediaan	Membuat Laporan	Untuk membuat laporan yang terkait dengan penerimaan dan permin-taan <i>spare part</i> .

D. Analisis Kebutuhan Non Fungsional

Berikut ini adalah hasil analisis kebutuhan nonfungsional yang sudah dikerjakan:

1. Ketersediaan (*Availability*), aplikasi hanya dapat diakses selama jam kerja bengkel dan tidak bisa diakses di luar jam kerja.

2. Kemudahan (*Portability*), aplikasi ini kompatibel dengan berbagai jenis *browser*.
3. Keamanan (*Safety*), sistem memiliki sistem *login* yang memerlukan pengguna untuk memasukkan *username* dan *password* sebelum dapat mengakses aplikasi.

E. Analisis Kebutuhan Sistem

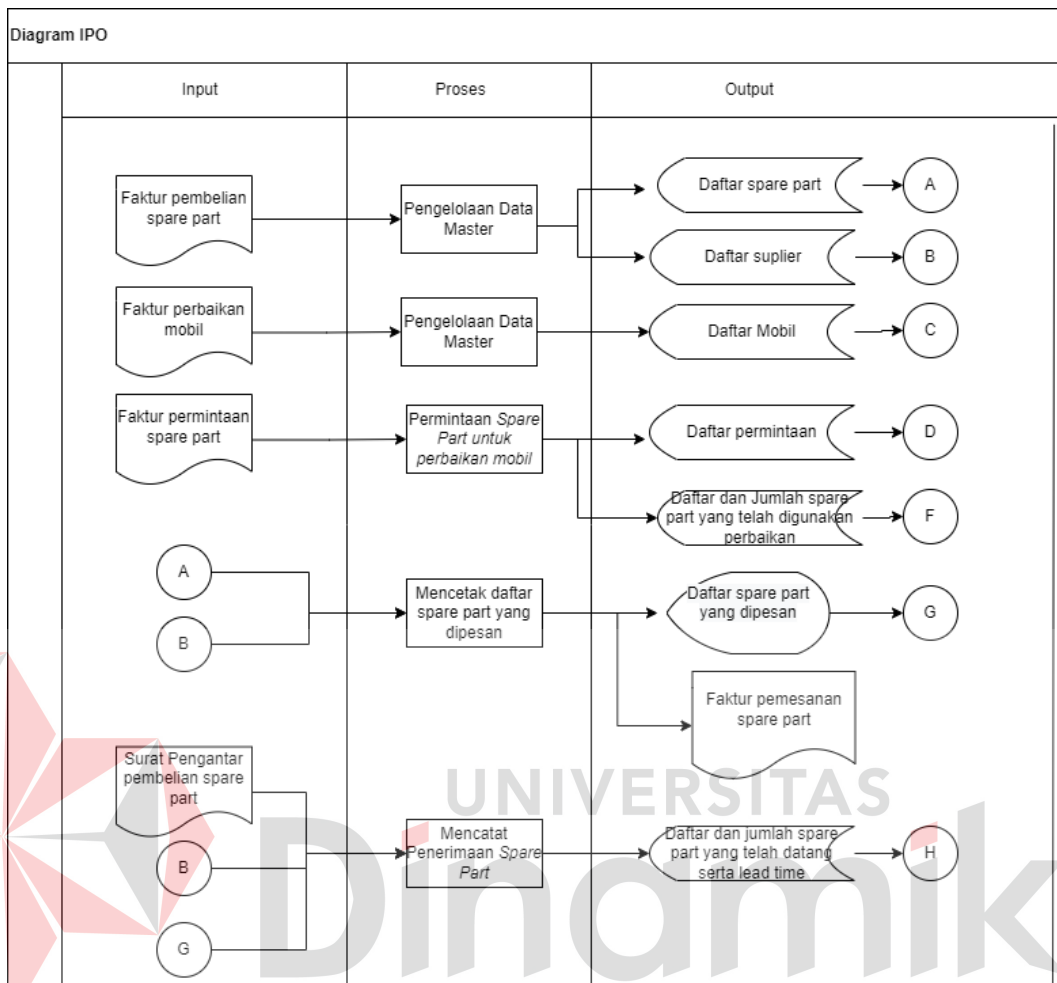
Tabel 3.5 merupakan hasil identifikasi peralatan yang diperlukan untuk mendukung arsitektur.

Tabel 3.5 Analisis Kebutuhan Sistem

Item	Kelompok	Spesifikasi
Server	Software	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>XAMPP</i> .
	Hardware	Prosesor minimum <i>Intel Core i3</i> atau lebih baru, memori minimum 2 GB atau lebih tinggi, dan kapasitas <i>hard disk</i> minimum 100 GB atau lebih tinggi.
Komputer Klien (Pengguna)	Software	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>web browser</i> .
	Hardware	Prosesor minimum <i>Intel Core i3</i> atau yang lebih baru, memori minimum 2 GB atau lebih tinggi, dan kapasitas <i>hard disk</i> minimum 150 GB atau lebih tinggi.
	Software	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>web browser</i> .
	Hardware	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>web browser</i> .
Komputer Pengembangan	Software	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>web browser</i> .
	Hardware	Sistem operasi <i>Windows 7 Service Pack 1</i> , <i>Windows 8.1</i> , <i>Windows 10</i> versi 64-bit, dan <i>web browser</i> .

F. Diagram IPO

Untuk menggambarkan alur input dan proses yang dibutuhkan untuk menghasilkan *output* sistem, menggunakan diagram IPO (*Input, Proses, Output*). Tujuannya adalah untuk memahami secara menyeluruh informasi yang diperlukan. Gambar 3.3 menggambarkan diagram IPO.



Gambar 3.3 Diagram IPO (1)

Untuk Diagram IPO selanjutnya mengenai proses pengendalian persediaan *spare part* yang bisa dilihat pada Gambar 3.4.

A. Input

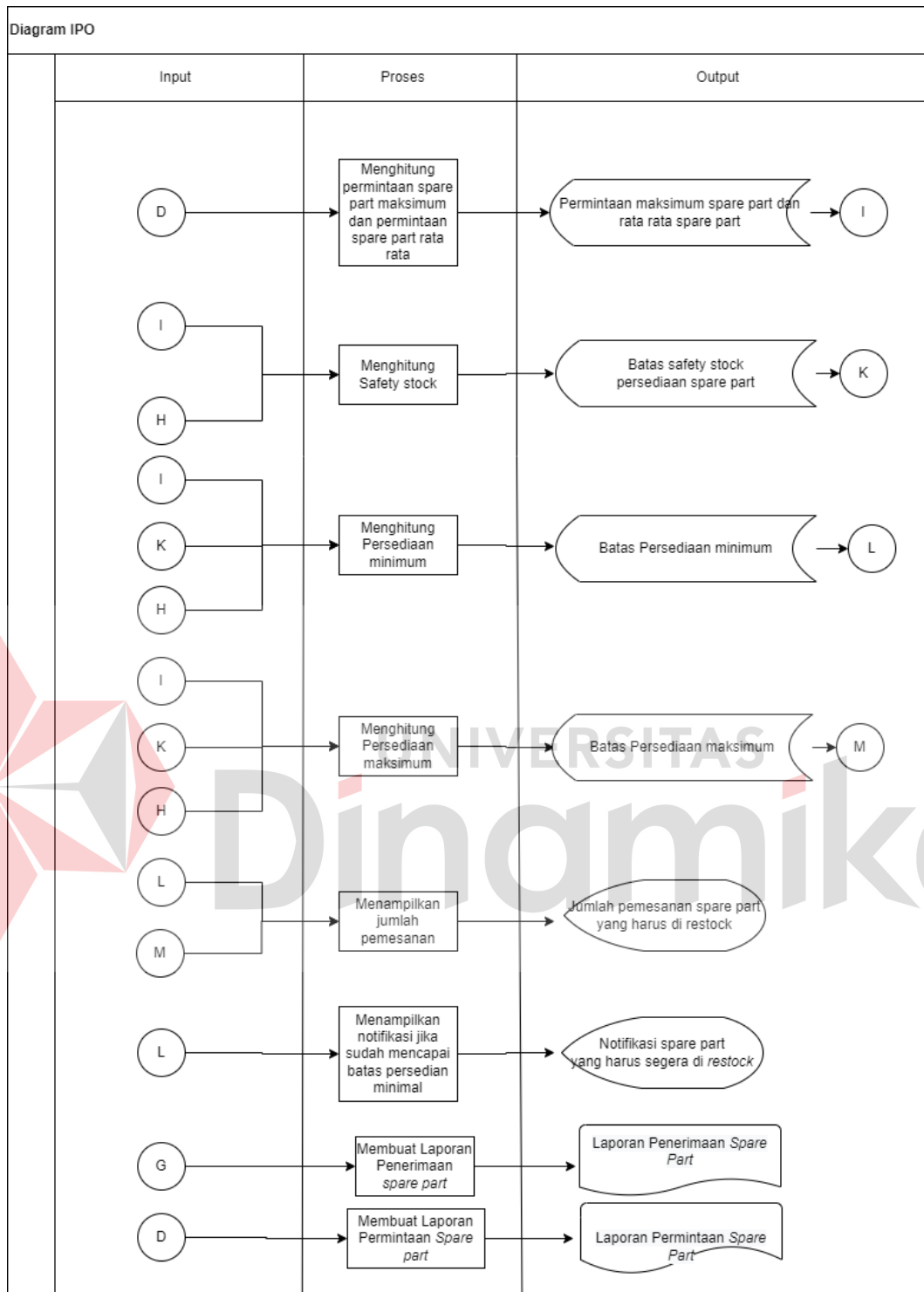
Aplikasi ini memiliki beberapa jenis *input*, termasuk:

a. Faktur pembelian *spare part*

Informasi yang dimasukkan dalam faktur spare part meliputi kode, nama *spare part*, stok, dan harga.

b. Faktur perbaikan mobil

Informasi yang dimasukkan dalam faktur perbaikan mobil meliputi kode, merk, tipe, jenis, dan tahun.



Gambar 3.4 Diagram IPO (2)

c. Faktor permintaan *spare part*

Daftar permintaan yang dimaksudkan berupa kode, kode *spare part*, kode mobil, nama *spare part*, nama mobil dan tanggal.

d. Surat pengantar pembelian *spare part*

Surat pengantar pembelian *spare part* berupa kode, kode *spare part*, kode daftar permintaan, nama *spare part*, dan tanggal.

B. Proses

Berdasarkan data yang diterima, data tersebut akan diproses untuk menghasilkan *output* yang dibutuhkan. Aplikasi ini melalui beberapa tahap proses, termasuk:

a. Pengelolaan *data master*

Pengelolaan data master melibatkan tiga elemen utama, yaitu *spare part*, mobil, dan *supplier*. Tugas utama dalam pengelolaan *data master* adalah menambahkan, memodifikasi, dan menghapus data yang terdapat dalam *database*.

b. Menerima permintaan *spare part* untuk perbaikan

Catatan permintaan *spare part* oleh administrasi bagian persediaan melibatkan beberapa tahap, termasuk menambahkan data permintaan. Data permintaan yang diterima kemudian digunakan dalam pengelolaan persediaan. Dalam hal ini, data permintaan digunakan untuk menghitung *safety stock*, termasuk rata-rata dan permintaan maksimum, yang dibutuhkan dalam perhitungan metode *Min Max*.

d. Mencetak daftar *spare part* yang di pesan

Mencatat dan mencetak daftar *spare part* yang dipesan adalah tahap pengelolaan data pemesanan *spare part*. Tahap ini melibatkan penambahan data ke dalam *database* dan proses mencetak daftar *spare part* yang dipesan.

e. Mencatat penerimaan *spare part*

Mencatat penerimaan *spare part* adalah tahap dalam proses pengelolaan data penerimaan *spare part*. Setelah bagian administrasi menerima *spare part* dari pemasok, akan melakukan proses catatan penerimaan. Hasil dari tahap ini akan digunakan untuk menentukan waktu pengiriman (*lead time*) dalam pengendalian persediaan.

f. Pengendalian persediaan metode *min max*

Proses pengendalian persediaan dengan menggunakan metode *min max* adalah salah satu cara untuk menentukan jumlah minimum dan maksimum dari

persediaan *spare part* yang harus tersedia. Dalam hal ini, diperlukan informasi permintaan dan waktu pengiriman (*lead time*) untuk melakukan perhitungan. Contoh simulasi perhitungan metode *min max* dapat ditemukan pada Tabel 4.6.

Tabel 3.6 Simulasi Min Max

Tanggal	Nama <i>spare part</i>	Jumlah permintaan (unit)
1	Evaporator	3
2	Evaporator	1
3	Evaporator	3
4	Evaporator	6
5	Evaporator	8
6	Evaporator	2
7	Evaporator	1
8	Evaporator	4
9	Evaporator	6
10	Evaporator	8
11	Evaporator	2
12	Evaporator	1
13	Evaporator	2
14	Evaporator	2
15	Evaporator	1
16	Evaporator	1
17	Evaporator	3
18	Evaporator	4
19	Evaporator	1
20	Evaporator	6
21	Evaporator	2
22	Evaporator	2
23	Evaporator	2
24	Evaporator	2
25	Evaporator	3
26	Evaporator	4
27	Evaporator	1
28	Evaporator	2
29	Evaporator	4
30	Evaporator	5
31	Evaporator	1

Berdasarkan simulasi tersebut diperoleh (untuk bulan Februari 2022) hasil sebagai berikut:

Total permintaan dalam 1 bulan = 93 unit

Permintaan rata-rata = 93 unit/hari : 30 hari = 3 unit/hari

Permintaan maksimum = 8 unit/hari

Lead Time = 9 hari

1. $Safety Stock = (\text{permintaan maksimum} - \text{permintaan rata-rata per periode}) \times L$

$$= (8 \text{ unit/hari} - 3 \text{ unit/hari}) \times 9 \text{ hari}$$

$$= 45 \text{ unit}$$

2. Persediaan minimum = (permintaan rata rata per periode x L) + SS

$$= (3 \text{ unit/hari} \times 9 \text{ hari}) + 45 \text{ unit}$$

$$= 72 \text{ unit}$$
3. Persediaan maksimum = 2 x (permintaan rata rata per periode x L) + SS

$$= 2 \times (3 \text{ unit/hari} \times 9 \text{ hari}) + 45 \text{ unit}$$

$$= 99 \text{ unit}$$
4. Jumlah pemesanan = persediaan maksimum - persediaan minimum

$$= 99 \text{ unit} - 72 \text{ unit}$$

$$= 27 \text{ unit}$$

g. Menampilkan notifikasi

Menampilkan notifikasi merupakan proses yang digunakan dapat mengetahui jumlah *spare part* yang harus di *restock*.

C) Output

Berikut adalah rekapitulasi hasil proses yang dilakukan oleh sistem, setelah *input* diterima dan diproses:

a. Daftar *spare part*

Informasi *spare part* yang tersimpan dalam *database* akan ditampilkan dalam daftar.

b. Daftar mobil

Data mobil akan ditampilkan pada daftar ini sebagai informasi.

c. Daftar *supplier*

Informasi tentang *supplier* yang tersimpan dalam sistem akan ditampilkan dalam daftar.

d. Daftar permintaan

Data permintaan *spare part* yang disimpan dalam *database* akan ditampilkan pada daftar ini. Informasi ini akan digunakan untuk menghitung jumlah minimal dan maksimal dari persediaan *spare part*.

- e. Jumlah dan daftar *spare part* yang telah digunakan perbaikan
Proses pengeluaran *spare part* oleh admin melibatkan beberapa tahap, mulai dari pembuatan daftar dan mencatat jumlah *spare part* yang dikeluarkan. Informasi yang dihasilkan dari proses ini akan digunakan untuk memantau dan memperbaharui jumlah persediaan *spare part*.
- f. Daftar pemesanan *spare part*
Informasi mengenai permintaan *spare part* yang diterima dari *supplier* melalui proses pemesanan.
- g. Faktur pemesanan *spare part*
Faktur pemesanan *spare part* menampilkan dokumen faktur pemesanan *spare part* dari *supplier*.
- h. Daftar dan jumlah *spare part* yang diterima dan *lead time*
Informasi mengenai daftar dan jumlah *spare part* yang telah diterima, serta informasi tentang *lead time*, digunakan untuk melakukan perhitungan pengendalian persediaan menggunakan metode *min max*.
- i. Permintaan maksimum *spare part* dan rata rata *spare part*
Informasi berupa nilai maksimum permintaan *spare part* dan rata rata *spare part*.
- j. Nilai minimal dan maksimal stok *spare part*
Informasi tentang jumlah minimum dan maksimum *spare part* yang dibutuhkan dapat diperoleh melalui proses pengendalian persediaan menggunakan metode *min max*. Hasil dari proses ini memberikan notifikasi bila jumlah persediaan *spare part* sudah hampir habis dan membutuhkan *restock*.
- k. Informasi jumlah *spare part* apabila mendekati nilai minimal
Informasi menunjukkan bahwa stok *spare part* harus segera diisi ulang untuk memastikan ketersediaan bahan.
- l. Mengetahui jumlah *spare part* yang harus *direct stock*
Hasil yang akan diterima oleh bagian administrasi menunjukkan daftar dan jumlah *spare part* yang perlu ditambahkan kembali ke stok.
- m. Laporan penerimaan *spare part*
Laporan transaksi penerimaan *spare part* akan menyediakan informasi yang diperlukan bagi bagian Administrasi persediaan, berupa daftar penerimaan *spare part* yang dilengkapi dengan informasi seperti jenis *spare part*, *supplier*, jumlah

yang dipesan, jumlah yang diterima, tanggal pemesanan, tanggal penerimaan, dan periode *lead time*. Laporan ini dapat dikeluarkan dalam bentuk tabel dan tersedia dalam format PDF.

n. Laporan permintaan *spare part*

Laporan permintaan spare part yang baru akan menunjukkan ringkasan transaksi permintaan spare part dalam bentuk tabel pdf. Isi laporan tersebut termasuk detail *spare part*, jumlah permintaan dan tanggal permintaan yang tercatat secara akurat.

3.2. Planning

Pada tahap *planning*, akan membuat jadwal untuk Proses pembangunan sistem akan dilaksanakan. Hasil akhir dari tahap ini adalah jadwal kerja yang dilampirkan dalam lampiran 1.

3.3. Modeling

Pada tahap *modeling*, akan dilakukan penggambaran desain aplikasi melalui beberapa tahap. Proses tersebut meliputi pembuatan *Sysflow* untuk menggambarkan alur proses bisnis dalam aplikasi, dan menggunakan aplikasi *draw.io*. Kemudian, akan dibuat *Data Flow Diagram (DFD)* yang bertujuan untuk menggambarkan aliran data dalam aplikasi menggunakan *Power Designer*. Setelah itu, akan dibuat *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)* juga menggunakan *Power Designer*. Pada tahap Desain *Interface*, penggambaran desain *interface*, *input*, dan *output* dari aplikasi akan dilakukan menggunakan aplikasi *Figma*.

3.3.1. System flow

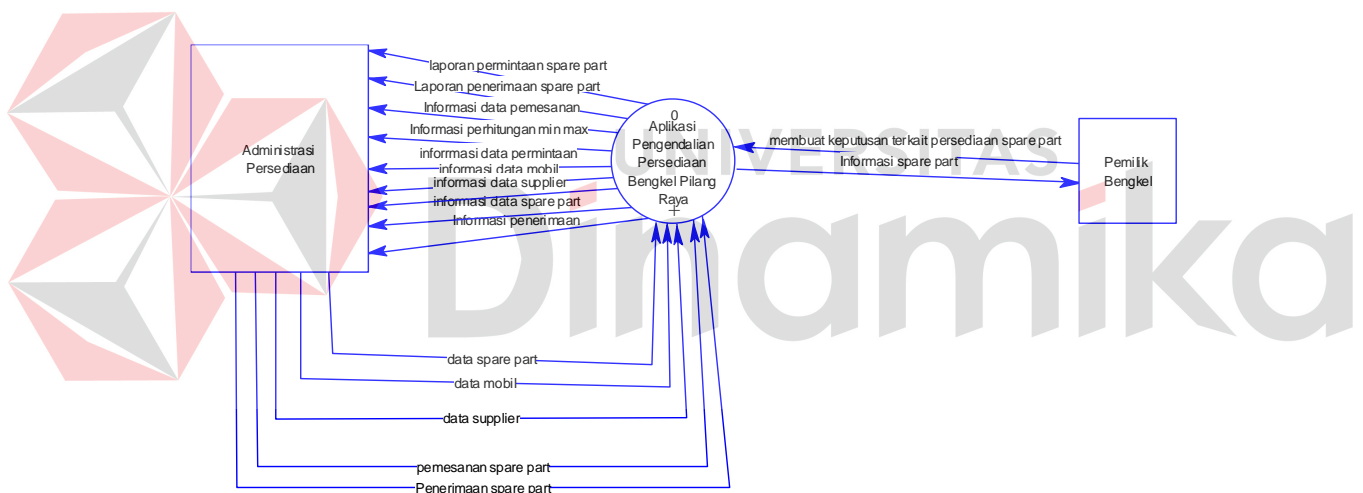
Dalam *system flow* aplikasi, *data master* dikelola dan diterapkan melalui 2 garis alir warna hitam dan biru. Garis alir biru terkait dengan tabel basis data dan garis alir hitam merupakan proses internal. Visualisasi dari sistem aliran aplikasi dapat ditemukan pada gambar 3.5.

3.3.2. Data Flow Diagram (DFD)

Dalam membuat *Data Flow Diagram*, tahapannya meliputi *context diagram*, DFD level 0, level 1, dan level 2. *Context diagram* akan menjelaskan secara singkat data yang diterima dan dikeluarkan oleh aktor-aktor dalam aplikasi. Tahap berikutnya, detail data akan diterangkan melalui proses-proses yang terkait pada level 0, dan seterusnya hingga level 2. Gambar yang menggambarkan *context diagram* dapat ditemukan pada Gambar 3.6.

3.3.3. Diagram Berjenjang

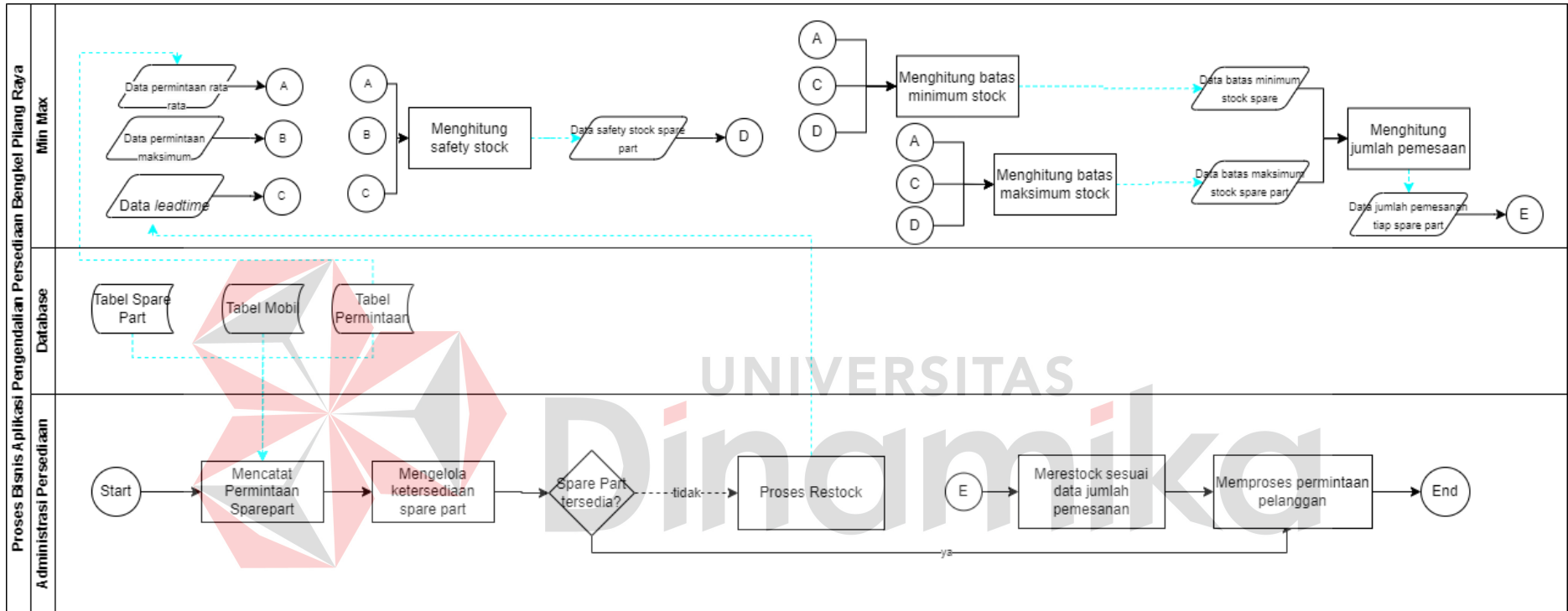
Diagram berjenjang berfungsi untuk menjelaskan seluruh proses yang terjadi pada aplikasi. Diagram berjenjang ini digunakan untuk mempersiapkan pembuatan DFD agar seluruh proses tercantum. Berikut adalah diagram berjenjang pada Gambar 3.7.



Gambar 3.5 Context diagram

3.3.4. Data flow diagram level 0

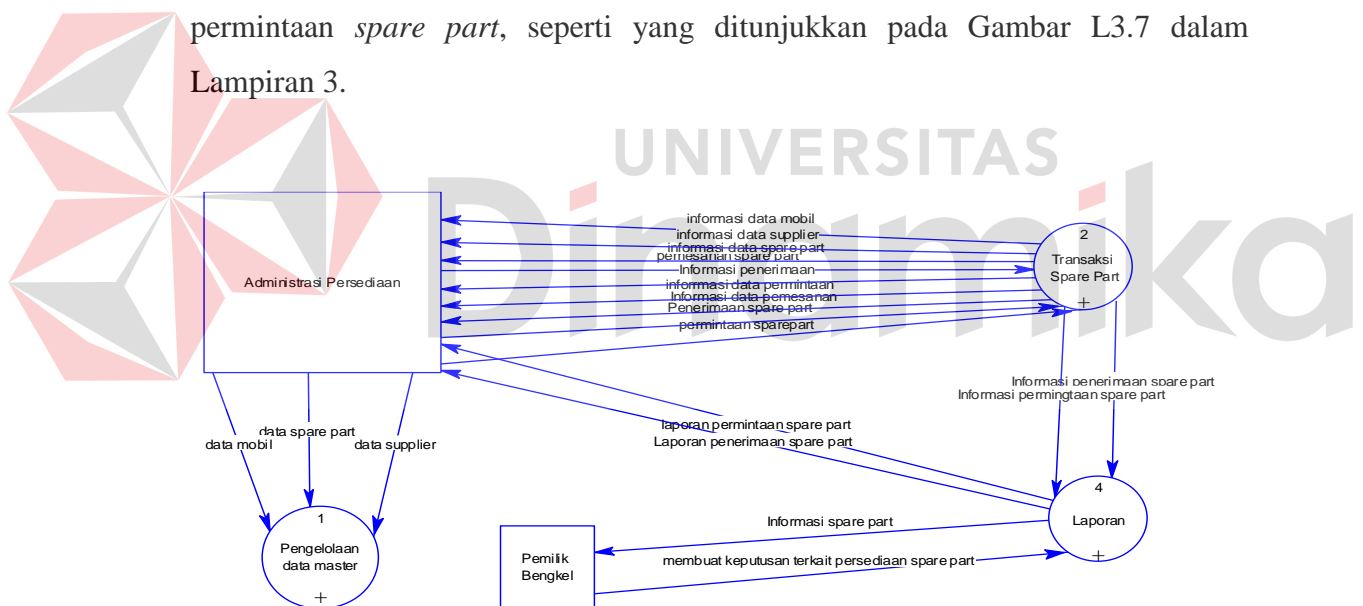
Pada tahap DFD 0, empat proses utama ditemukan. Struktur DFD tingkat 0 terdiri dari pengelolaan *data master*, transaksi *spare part*, perhitungan *min-max*, dan laporan. Namun, tabel tidak termasuk dalam diagram DFD tingkat 0 karena banyaknya proses yang membuat diagram sulit dibaca. Informasi lebih rinci dapat ditemukan pada Gambar 3.8 yang menunjukkan DFD tingkat 0.



Gambar 3.6 System Flow Diagram

3.3.5. Data flow diagram level 1

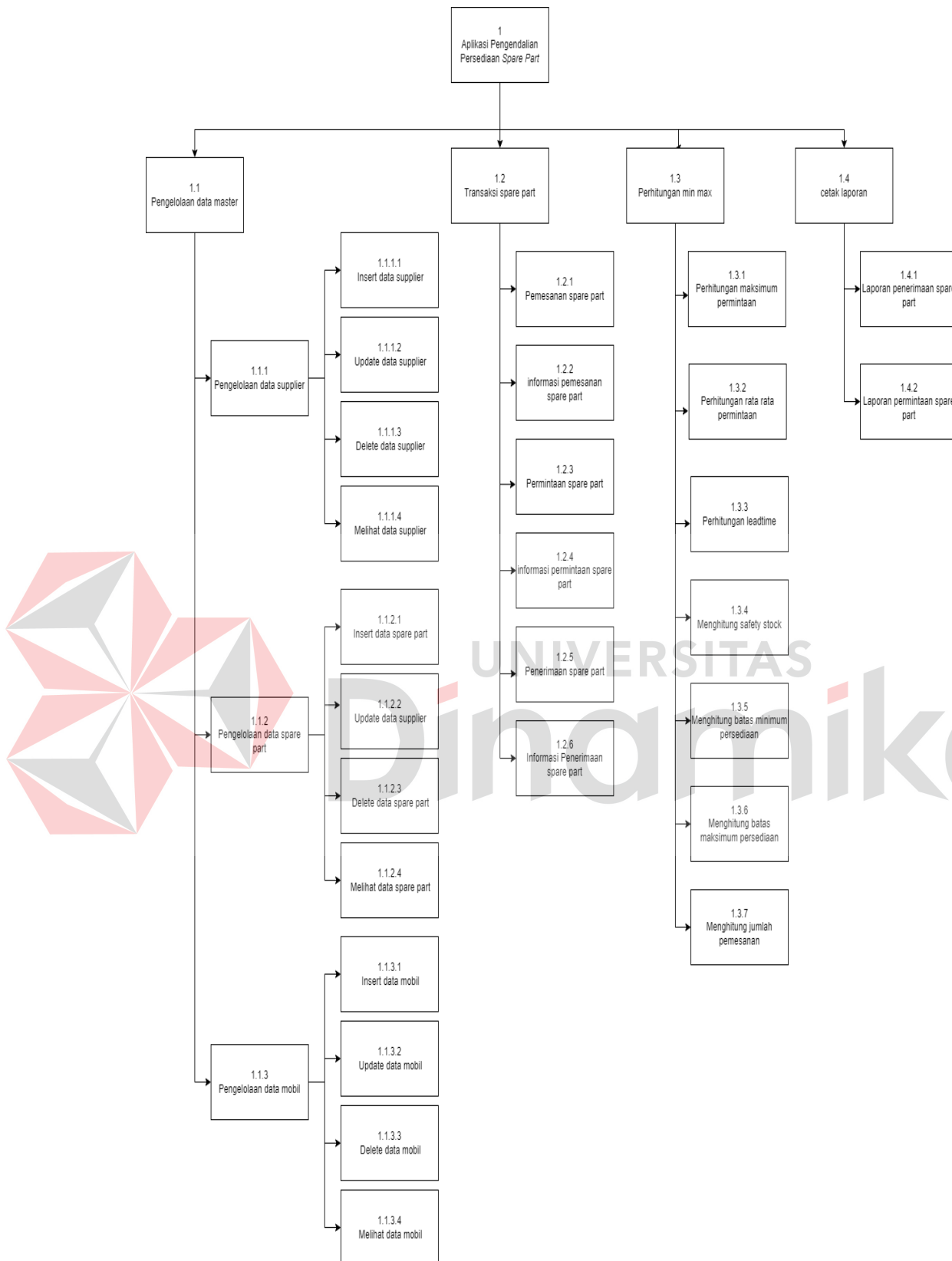
Pada tahap pemodelan DFD level 1, setiap proses dalam level 0 akan diterangkan dengan lebih rinci. Ada empat proses utama yaitu pengelolaan *data master*, transaksi *spare part*, perhitungan *min-max*, dan pembuatan laporan. Masing-masing proses memiliki diagram sendiri yang dapat ditemukan dalam lampiran. Proses pengelolaan data master meliputi pengelolaan data mobil, *spare part*, dan *supplier*, seperti yang ditunjukkan pada Gambar L3.1 dalam Lampiran 3. Proses transaksi *spare part* meliputi penerimaan, permintaan, dan pemesanan *spare part*, seperti yang ditunjukkan pada Gambar L3.5 dalam Lampiran 3. Proses perhitungan *min-max* mencakup perhitungan nilai minimum dan maksimum dari jumlah *spare part* yang ada, seperti yang ditunjukkan pada Gambar L3.6 dalam Lampiran 3. Proses laporan meliputi pembuatan laporan penerimaan dan permintaan *spare part*, seperti yang ditunjukkan pada Gambar L3.7 dalam Lampiran 3.



Gambar 3.7 data flow diagram level 0

3.3.6. Data flow diagram level 2

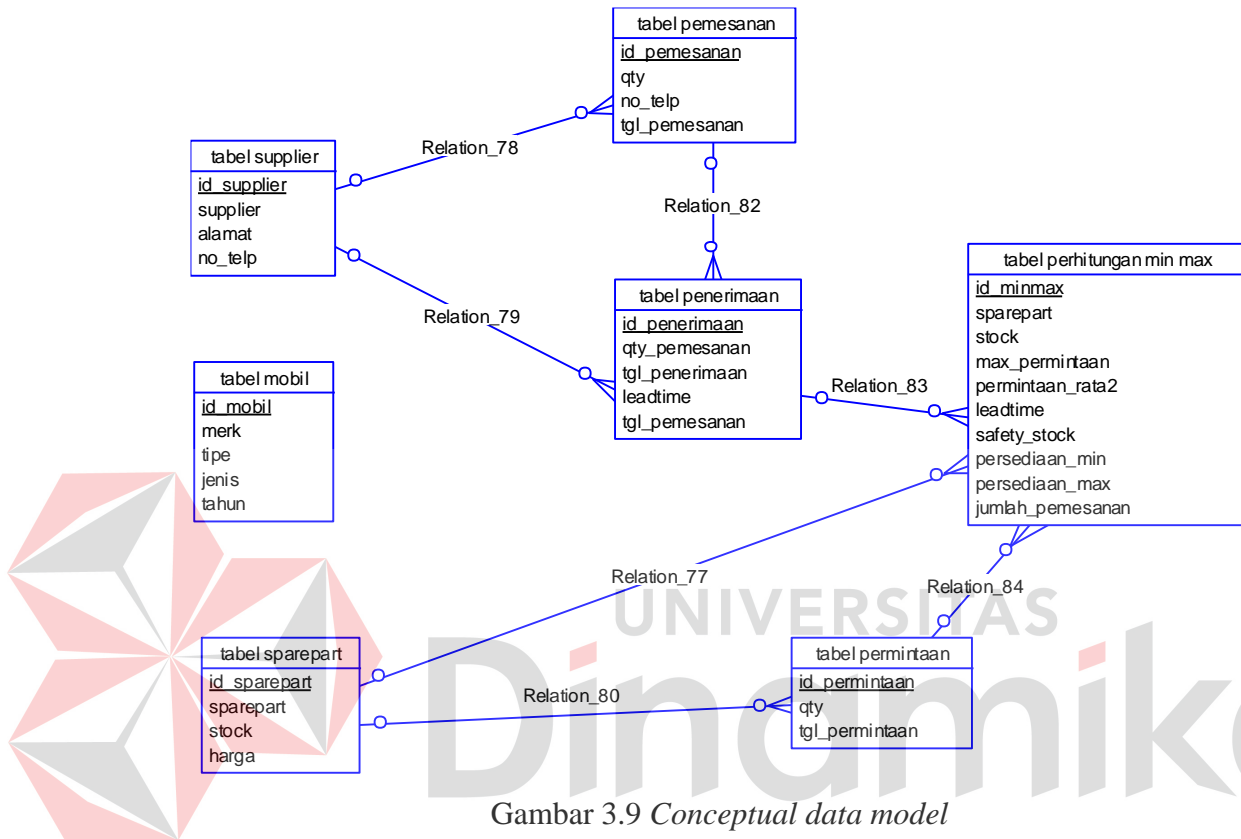
Pada tingkat DFD level 2, masing-masing proses pada DFD level 1 pengelolaan *data master* akan diperincikan. Detail-detail ini meliputi bagaimana data master *spare part*, mobil, dan *supplier* dikelola, dan dapat ditemukan pada Lampiran 3 yang berisi Diagram Alir Data.



Gambar 3.8 Diagram berjenjang

3.3.7 Conceptual data model

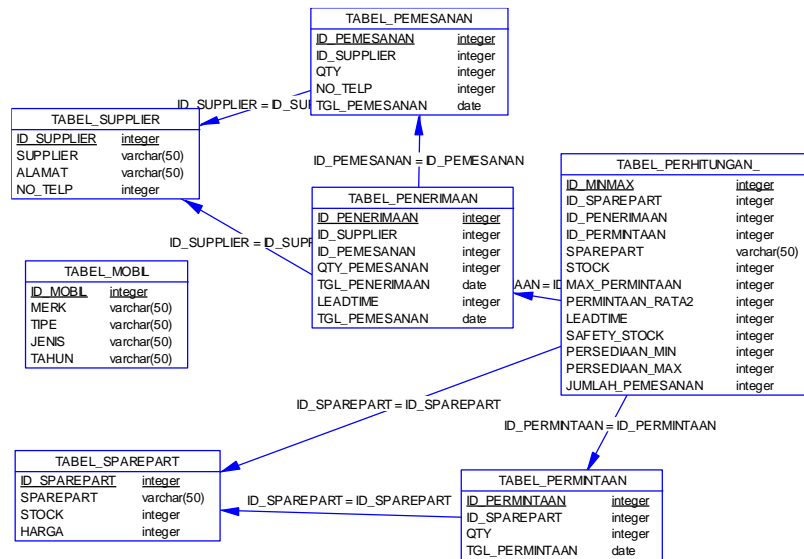
Berikut ini merupakan model data konseptual untuk aplikasi yang akan dibuat. Gambar model data konseptual dapat ditemukan pada Gambar 3.9.



Gambar 3.9 Conceptual data model

3.3.8 Physical data model

Berikut adalah *physical data model* untuk aplikasi yang dibuat. Gambar *physical data model* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Physical data model

3.3.9 Desain Interface

Desain *interface* pengendalian persediaan beranda akan memaparkan informasi mengenai jumlah barang, daftar transaksi yang terdiri dari permintaan, pemesanan, dan penerimaan, serta laporan yang berkaitan. Desain antarmuka tersebut bisa dilihat pada Lampiran 4.

3.3.9 Desain Input

Desain *input* pengendalian persediaan faktur pembelian *spare part*, faktur perbaikan mobil, faktur permintaan *spare part* dan surat pengantar pembelian *spare part* bisa dilihat pada Lampiran 5.

3.3.10 Desain Output

Desain *ouput* pengendalian persediaan daftar *spare part* yang dipesan dan faktur pemesanan *spare part* dapat dilihat pada Lampiran 6.

3.4. Construction

Tahap ini dilakukan proses pembangunan dan pembuatan aplikasi dengan cara mengkodekan (*Coding*) berdasarkan desain yang sudah direncanakan.

3.4.1. Coding

Pada tahap *construction*, akan dilakukan proses pengkodean untuk menghasilkan aplikasi yang sesuai dengan desain yang sudah ditentukan. Alat yang akan digunakan dalam proses pengkodean adalah *Visual Studio Code*, *XAMPP*, bahasa pemrograman *PHP*, *HTML*, *CSS*, *JavaScript*, dan *database MySQL*. Dalam tahap ini akan diterapkan metodologi pengendalian persediaan *min max*.

3.4.2. Testing

Pada tahap *testing*, akan melakukan *Black Box Testing* dan *User Acceptance Testing (UAT)*. *Black Box Testing* adalah tahap di mana aplikasi diuji oleh *end user* untuk memastikan bahwa tidak ada *error* pada *input*, proses, dan *output* dari aplikasi. Setelah itu, melanjutkan ke *User Acceptance Testing (UAT)*. Pada tahap ini, aplikasi dijalankan oleh pengguna dan diberikan kuisioner mengenai apakah fungsi aplikasi sudah sesuai atau tidak. Skenario untuk *Black Box Testing* dapat dilihat pada Tabel 3.7.

Tabel 3.7 Skenario *Black Box Testing Login*

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

Selanjutnya pada Administrasi Persediaan terdapat Skenario untuk *Black Box Testing* saat menjalankan aplikasi yang dapat dilihat pada Tabel 3.2

Tabel 3.8 Skenario *Black Box Testing Administrasi Persediaan*

No	Skenario
Permintaan Spare Part (Memastikan pengguna dapat mengelola permintaan spare part)	
1	Melakukan <i>input permintaan spare part</i> .
2	Tidak melakukan <i>input permintaan spare part</i> .
3	Mengubah data <i>permintaan spare part</i>
4	Melakukan hapus <i>permintaan spare part</i>
Melihat Data Spare Part (Memastikan pengguna dapat melihat data spare part)	
5	Melakukan melihat data <i>spare part</i>

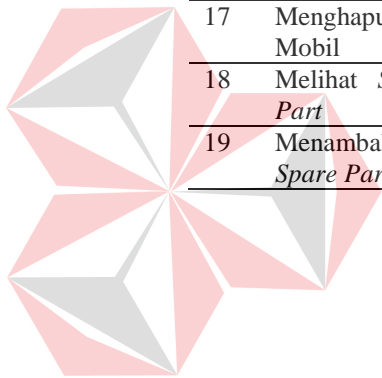
No	Skenario
Pengelolaan Data Master (Memastikan pengguna dapat mengelola data master)	
6	Melakukan <i>input</i> data <i>supplier</i>
7	Melakukan <i>edit</i> data <i>supplier</i>
8	Melakukan hapus data <i>supplier</i>
9	Melakukan <i>input</i> data mobil
10	Melakukan <i>edit</i> data mobil
11	Melakukan hapus data mobil
12	Melakukan <i>input</i> data <i>sparepart</i>
13	Melakukan <i>edit</i> data <i>sparepart</i>
14	Melakukan hapus data <i>sparepart</i>
15	Memasukkan data pemesanan <i>spare part</i>
16	Melakukan ubah data pemesanan <i>spare part</i>
17	Melakukan hapus data pemesanan <i>spare part</i>
Laporan (Memastikan pengguna dapat mencetak laporan)	
18	Melakukan print laporan permintaan <i>spare part</i>
19	Melakukan print laporan penerimaan <i>spare part</i>

Sementara pada Administrasi Persediaan terdapat skenario untuk UAT yang dapat dilihat pada Tabel 3.3.

Tabel 3.9 Skenario UAT Administrasi Persediaan

No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
1	<i>Login</i>	Jika pengguna berhasil <i>login</i> di dalam aplikasi, maka diarahkan ke beranda. Namun, jika <i>login</i> tidak berhasil, muncul pesan bahwa "username atau password tidak sesuai"			
2	Beranda	Pengguna dapat melihat jumlah dan daftar permintaan <i>spare part</i> , daftar penerimaan <i>spare part</i> , dan daftar pemesanan <i>spare part</i>			
3	Pengendalian Persediaan	Pengguna dapat melihat daftar dan jumlah perhitungan <i>min max</i> serta daftar dan jumlah <i>spare part</i> yang harus disiapkan			
4	Melihat Permintaan	Permintaan <i>spare part</i> bisa dilihat pengguna			
5	Menambah Pemesanan	Pengguna dapat menambahkan pemesanan. Jika penambahan berhasil, muncul pesan "data berhasil ditambahkan"			
6	Melihat Penerimaan	Penerimaan <i>spare part</i> bisa dilihat pengguna			
7	Konfirmasi Penerimaan	Pengguna dapat konfirmasi penerimaan, apabila berhasil akan muncul "data penerimaan berhasil ditambahkan"			

No	Test Case	Deskripsi	Test Result		Comments
			Accept	Reject	
8.	Print Laporan permintaan <i>spare part</i>	Pengguna dapat mencetak laporan permintaan <i>spare part</i> dalam bentuk file pdf.			
9	Print Laporan Penerimaan <i>Spare Part</i>	Pengguna dapat mencetak laporan penerimaan <i>spare part</i> dalam bentuk file pdf.			
10	Melihat <i>Supplier</i>	Pengguna melihat data <i>supplier</i> .			
11	Menambah <i>Supplier</i>	Pengguna menambahkan data <i>supplier</i> .			
12	Mengubah <i>Supplier</i>	Pengguna mengubah data <i>supplier</i> .			
13	Menghapus <i>Supplier</i>	Pengguna dapat menghapus data <i>supplier</i> .			
14	Melihat Mobil	Pengguna melihat data mobil			
15	Menambah Mobil	Pengguna menambahkan data mobil			
16	Mengubah Mobil	Pengguna mengubah data mobil			
17	Menghapus Mobil	Pengguna menghapus data mobil			
18	Melihat <i>Spare Part</i>	Pengguna melihat data <i>Spare part</i>			
19	Menambah <i>Spare Part</i>	Pengguna menambahkan data <i>spare part</i>			



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada tahap implementasi, rancangan yang dibuat diterapkan dalam bentuk aplikasi atau situs *web*. Setelah aplikasi atau situs *web* dibuat, dilakukan evaluasi untuk mengetahui efektivitas dari implementasi tersebut melalui pengujian (*testing*).

4.2 Implementasi Sistem

Tahapan implementasi sistem adalah konsekuensi dari proses identifikasi, analisis, dan desain yang dilakukan sebelumnya. Dalam tahapan ini, sistem diterapkan dengan melakukan pengkodean untuk memastikan bahwa aplikasi atau *website* dapat berfungsi sesuai dengan rancangan. Berikut adalah hasil dari implementasi sistem untuk Administrasi Persediaan yang menggunakan desain terbaru dan teroptimalkan.

A. Login

Pada bagian *login*, pengguna diminta untuk memasukkan identitas akses aplikasi, yaitu *username* dan *password*. Jika informasi *login* yang dimasukkan benar, pengguna akan dialihkan ke halaman utama aplikasi. Namun, jika informasi *login* yang dimasukkan salah, akan muncul notifikasi kesalahan yang mengindikasikan bahwa *username* atau *password* yang dimasukkan tidak sesuai. Implementasi dari proses *login* bisa ditemukan pada Lampiran 7

B. Melihat Data *Spare Part*

Pada fungsi ini, pengguna dapat melihat informasi tentang daftar spare part yang tersedia, termasuk kode *spare part*, nama *spare part*, harga, dan jumlah stok. Detail implementasi dapat ditemukan pada Lampiran 7.

C. Proses Permintaan *Spare Part*

Pada fungsi ini, Adminiistrasi Persediaan dapat melihat data permintaan dan menambahkan data permintaan. Detail implementasi dapat ditemukan pada Lampiran 7.

D. Pengelolaan Data Master

Pada fungsi ini, Administrasi Persediaan memiliki kemampuan untuk mengelola *data master* yang terdiri dari *supplier*, mobil, dan *spare part*. Termasuk dalam kemampuan ini adalah melakukan operasi seperti *input*, *edit*, dan *delete* data. Detail implementasi dapat ditemukan pada Lampiran 7.

E. Pemesanan Spare Part

Pada fungsi ini, Administrasi Persediaan memiliki fitur membuat pemesanan *spare part*. Detail implementasi dapat ditemukan pada Lampiran 7.

F. Konfirmasi Penerimaan Spare Part

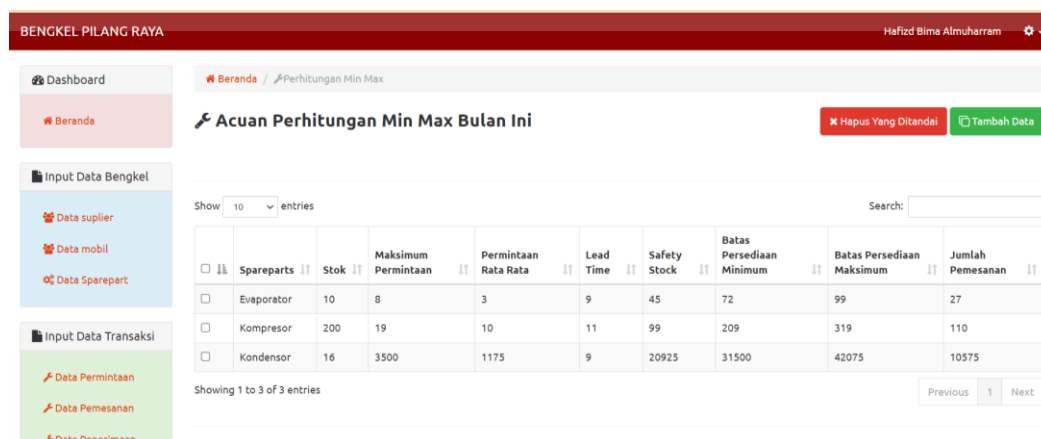
Pada fungsi ini, pengguna melakukan konfirmasi penerimaan *spare part*. Detail implementasi dapat ditemukan pada Lampiran 7.

H. Laporan

Pada fungsi ini pengguna melakukan mencetak laporan permintaan *spare part* dan penerimaan *spare part*. Detail implementasi dapat ditemukan pada Lampiran 7.

I. Melihat Hasil Perhitungan *Min Max*

Pada fitur ini, pengguna dapat memantau hasil perhitungan metode *Min Max*, daftar dan jumlah *spare part* yang perlu diperbaharui dalam persediaan. Implementasinya dapat ditemukan pada Gambar 4.1



The screenshot shows a web application interface for 'BENGKEL PILANG RAYA'. The main content area is titled 'Acuan Perhitungan Min Max Bulan Ini'. It features a table with the following data:

Spareparts	Stok	Maksimum Permintaan	Permintaan Rata Rata	Lead Time	Safety Stock	Batas Persediaan Minimum	Batas Persediaan Maksimum	Jumlah Pemesanan
Evaporator	10	8	3	9	45	72	99	27
Kompresor	200	19	10	11	99	209	319	110
Kondensor	16	3500	1175	9	20925	31500	42075	10575

Gambar 4.1 Implementasi Hasil Perhitungan *Min Max*

4.3. Hasil *Testing*

Hasil *testing* terfokus pada pengendalian persediaan dengan hasil pengujian atau implementasi. Berikut adalah hasil pengujian yang telah dilakukan dengan pengujian *Black Box Testing* dan *User Acceptance Testing*

4.3.1. *Black Box Testing*

Berikut adalah hasil uji aplikasi menggunakan metode *Black Box Testing* yang sesuai dengan skenario yang ditetapkan, tertera pada Tabel 4.1. Informasi lebih lanjut dapat ditemukan pada Lampiran 8.

Tabel 4.1 Hasil *Black Box Testing*

No	Fungsi	Persentase Keberhasilan
1	<i>Login</i>	100%
2	Melakukan permintaan <i>spare part</i>	100%
3	Melakukan melihat data <i>spare part</i>	100%
4	Melakukan pengelolaan <i>data master</i>	100%
5	Melakukan pemesanan <i>spare part</i>	100%
6	Melakukan penerimaan <i>spare part</i>	100%
7	Melakukan menceta laporan	100%
8	Melakukan melihat hasil perhitungan <i>min max</i>	100%

4.3.2. *User Acceptance Testing (UAT)*

UAT dilakukan pada Administrasi Persediaan dan Pemilik Bengkel di bengkel Pilang Raya, dengan pengguna menjalankan aplikasi sesuai dengan skenario dan mengisi kuisioner yang diberikan. Bobot Kriteria UAT dilihat pada Tabel 4.2.

Tabel 4.2 Kriteria UAT

Jawaban	Keterangan	Bobot
A	Sangat Mudah	5
B	Mudah	4
C	Netral	3
D	Kurang Bagus	2
E	Sangat Sulit	1

Berikut ini adalah tabel hasil dari *User Acceptance Testing* yang dilakukan pada administrasi persediaan, tertera pada Tabel 4.3. Informasi lebih lanjut tentang hasil UAT dapat ditemukan pada Lampiran 9.

Tabel 4.3 Hasil Responden UAT Adminstrasi Persediaan

Kode Pertanyaan	Nilai (Bobot x Jumlah Jawaban)					Jumlah	Persentase Penerimaan Aplikasi
	A	B	C	D	E		
1	1	0	0	0	0	1	100%
2	1	0	0	0	0	1	100%
3	1	0	0	0	0	1	100%
4	1	0	0	0	0	1	100%
5	1	0	0	0	0	1	100%
6	1	0	0	0	0	1	100%
7	1	0	0	0	0	1	100%
8	1	0	0	0	0	1	100%
9	1	0	0	0	0	1	100%
10	1	0	0	0	0	1	100%
11	1	0	0	0	0	1	100%
12	1	0	0	0	0	1	100%
13	1	0	0	0	0	1	100%
14	1	0	0	0	0	1	100%
15	1	0	0	0	0	1	100%
16	1	0	0	0	0	1	100%
17	1	0	0	0	0	1	100%
18	1	0	0	0	0	1	100%
19	1	0	0	0	0	1	100%

Berikut merupakan tabel hasil responden *User Acceptance Testing* terhadap pertanyaan yang diberikan kepada pemilik bengkel, yang bisa dilihat Tabel 4.4.

Tabel 4.4 Hasil Responden UAT Pemilik Bengkel

Kode Pertanyaan	Nilai (Bobot x Jumlah Jawaban)					Jumlah	Persentase Penerimaan Aplikasi
	A	B	C	D	E		
1	1	0	0	0	0	1	100%

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan analisis yang dilakukan pada bab-bab sebelumnya, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi ini memiliki fitur-fitur seperti: pengelolaan *data master* yang mencakup data *spare part*, *supplier*, dan mobil, transaksi *spare part* yang meliputi permintaan *spare part*, pemesanan *spare part*, dan penerimaan *spare part*, perhitungan metode *Min Max*, serta pembuatan laporan yang menyajikan laporan permintaan dan penerimaan *spare part*.
2. Aplikasi telah diuji dengan *black box testing*, yang hasilnya 100% semua fitur sudah beroperasi dengan sukses dan hasil yang memuaskan.
3. Aplikasi juga diuji dengan UAT, yang hasilnya semua responden sangat setuju artinya aplikasi sesuai dengan kebutuhan pengguna.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan sebelumnya, disarankan agar:

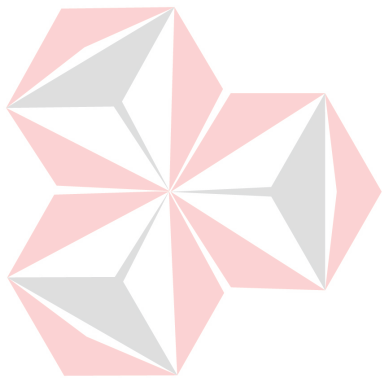
1. Menambahkan modul pengendalian persediaan yang lain untuk menyempurnakan aplikasi pengendalian persediaan bengkel.
2. Menambahkan akses penggunaan dengan berbasis *mobile* agar memudahkan pengguna mengakses aplikasi pengendalian persediaan bengkel.

DAFTAR PUSTAKA

- Hamzah, M. J. (2022). *Rancang Bangun Aplikasi Pengendalian Barang Berbasis Website dengan Metode Reorder Point Pada CV Djaja Aksa Mandiri*. Surabaya: Universitas Dinamika.
- Harminingtyas, R. (2014). ANALISIS LAYANAN WEBSITE SEBAGAI MEDIA PROMOSI, MEDIA TRANSAKSI DAN MEDIA INFORMASI DAN PENGARUHNYA TERHADAP BRAND IMAGE PERUSAHAAN PADA HOTEL CIPUTRA DI KOTA SEMARANG. *JURNAL STIE SEMARANG*, 37-57.
- Haryono, K., & Rahayu, N. (2019). User Acceptance Testing (UAT) Pada Purwarupa Sistem Tabungan Santri (Studi Kasus: Pondok Pesantren AlMawaddah). *Jurnal Ilmiah Multimedia dan Komunikasi, Volume 5 Nomor 1*, 1-10.
- Hertanto, R. H. (2020). METODE MIN-MAX DAN PENERAPANNYA SEBAGAI PENGENDALI PERSEDIAAN BAHAN BAKU PADA PT. BALATIF MALANG. *Jurnal Administrasi dan Bisnis*, 161-167.
- Lazuardi, M. L. (2019). Design Thinking David Kelley & Tim Brown: Otak Dibalik Penciptaan. *Jurnal Saintifik Manajemen dan Akuntansi*, 1-11.
- Leidiyana, H., & Anugrah, A. (2020). Aplikasi Pengendalian Persediaan Barang Berbasis Android dengan Metode Economic Order Quantity (EOQ) pada Bengkel Dunia Motor. *Jurnal Komtika*, 51-58.
- Pressman, R. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku 1*. Yogyakarta: ANDI.
- Putra, I. G. (2021). Sistem Informasi Manajemen Bengkel Modul Point of Sales Berbasis Web. *Modul Point of Sales Berbasis Web*, 1-12.
- Putri, D. M., & Ulkhaq, M. M. (2017). ANALISIS PENGENDALIAN PERSEDIAAN BAHAN BAKU KERTAS DUPLEX 120 GRAM DENGAN METODE MIN-MAX SYSTEM DI PT. JAYA AFLAHA, BATAM. *4th Annual Conference in Industrial and System Engineering* (pp. 412-418). Semarang: 4th Annual Conference in Industrial and System Engineering .
- Rahmanila, N. (2014, Februari 14). *Apa Itu Black Box Testing? Yuk, Kenali Arti, Manfaat, dan Jenis-jenisnya*. Retrieved from Retrieved from Glints: <https://glints.com/id/lowongan/black-box-testing/#.YUvbS7gzaMp>
- Sayang, E. S., Setyawan, H. B., & Lemantara, J. (2022). Rancang Bangun Aplikasi Pengendalian Persediaan Bahan Baku Berbasis Website Menggunakan

Metode Safety Stock dengan Perhitungan Service Level pada CV Dwi Teknik. *JSIKA*, 61-72.

Wahid, A., & Munir, M. (2020). Economic Order Quantity Istimewa pada Industri Krupuk “Istimewa” Bangil. *Journal of Industrial View*, 1-8.



UNIVERSITAS
Dinamika