



**RANCANG BANGUN APLIKSI PENGELOIAAN KATALOG SPAREPART
SEPEDA MOTOR BERBASIS WEBSITE PADA BENGKEL LENA GARAGE**

KERJA PRAKTIK



Oleh:

ARYSATYA HADI

22410100046

FAKULTAS TEKNOLOGI DAN INFORMATIKA

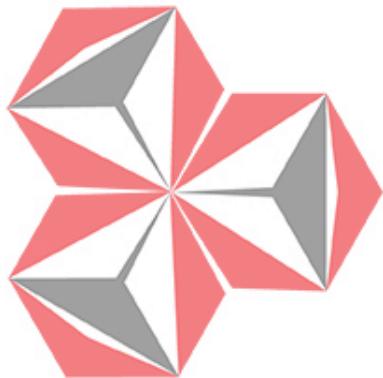
UNIVERSITAS DINAMIKA

2026

**RANCANG BANGUN APLIKSI PENGELOIAAN KATALOG
SPAREPART SEPEDA MOTOR BERBASIS WEBSITE PADA
BENGKEL LENA GARAGE**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana



UNIVERSITAS
Dinamika
Disusun Oleh:
Nama : Arysatya Hadi

NIM : 22410100046

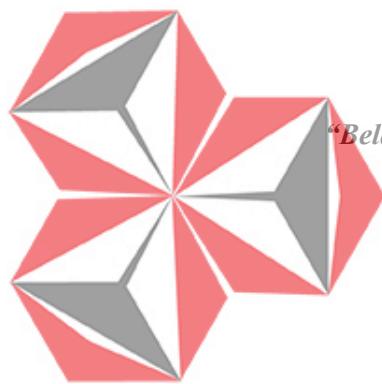
Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2026



“Belajar, beradaptasi, dan bertanggung jawab atas setiap hasil”

UNIVERSITAS
Dinamika

LEMBAR PENGESAHAN

Rancang Bangun Aplikasi Pengelolaan Katalog Sparepart Sepeda Motor

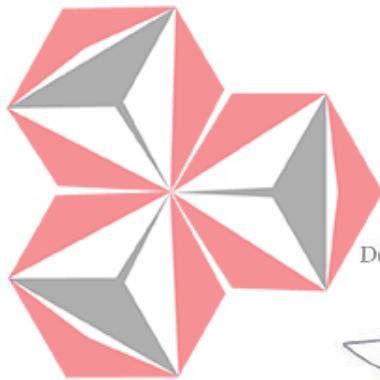
Berbasis *Website* Pada Bengkel Lena Garage

Laporan Kerja Praktik

oleh:

Arysatya Hadi

NIM. 22410100046



Telah diperiksa, diuji, dan disetujui

Dosen Pembimbing

Sulistiwati, S.Si., M.M

NIDN. 0719016801

UNIVERSITAS
Surabaya, 05 Januari 2026
Disetujui
Penyelia,

Lena Garage

Megantara Narendra P

Mengetahui,

Ketua Program Studi S1 Sistem Informasi

Digitally signed by
Endra Rahmawati
Date: 2026.01.23
08:36:34 +07'00'

Endra Rahmawati, M.Kom.

NIDN. 0712108701

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : Arysatya Hadi
NIM : 22410100046
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Kerja Praktik
Judul Karya : RANCANG BANGUN APLIKSI PENGELOIAAN KATALOG SPAREPART SEPEDA MOTOR BERBASIS WEBSITE PADA BENGKEL LENA GARAGE

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 Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap meneantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikan surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 28 Januari 2026



Arysatya hadi
NIM : 22410100046

ABSTRAK

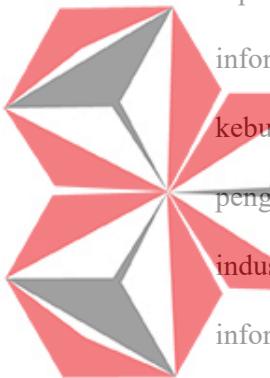
Kerja Praktik ini dilaksanakan di Lena Garage yang bergerak di bidang jasa perawatan dan perbaikan kendaraan. Tujuan dari pelaksanaan Kerja Praktik ini adalah untuk menerapkan ilmu pengetahuan yang telah diperoleh selama perkuliahan ke dalam dunia kerja serta memahami proses kerja secara langsung di lingkungan industri. Selama pelaksanaan Kerja Praktik, penulis terlibat dalam pengembangan dan pengelolaan *website* berbasis PHP dan MySQL yang berfungsi sebagai media pengolahan *data* dan penyampaian *informasi*. *Website* tersebut dibangun menggunakan HTML sebagai struktur tampilan, PHP sebagai bahasa pemrograman server-side, dan MySQL sebagai basis *data*, dengan dukungan XAMPP sebagai server lokal serta Visual Studio Code sebagai *text editor*. Hasil dari Kerja Praktik ini menunjukkan bahwa pemanfaatan teknologi *website* dapat membantu meningkatkan efisiensi pengelolaan *data* dan mendukung aktivitas operasional di Lena Garage. Diharapkan laporan ini dapat memberikan manfaat bagi perusahaan dan menjadi referensi bagi pengembangan sistem *informasi* di masa mendatang.

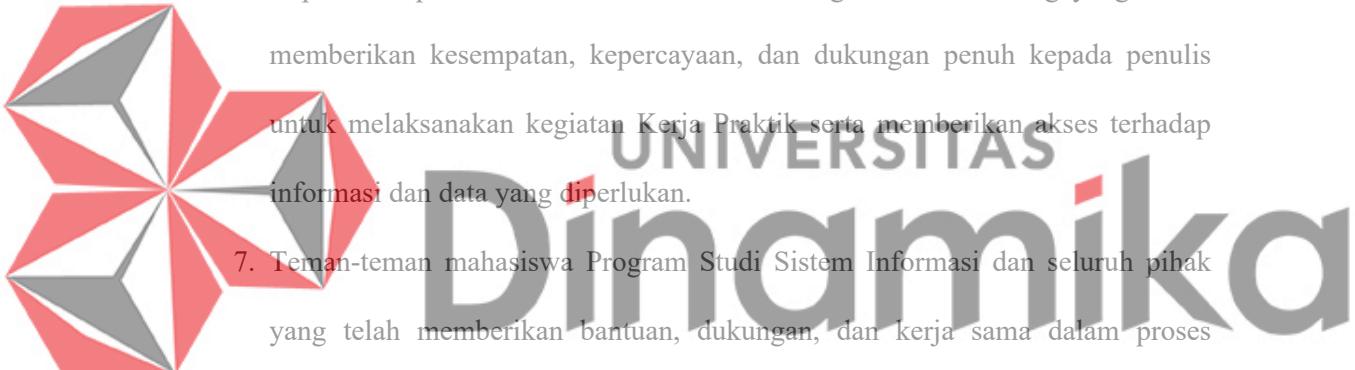
Kata Kunci: Kerja Praktik, Lena Garage, *Website*, PHP, MySQL, Sistem *Informasi*, Pengelolaan *Data*, Jasa Perawatan Kendaraan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat, berkah, dan karunia-Nya sehingga penulis dapat menyelesaikan laporan Kerja Praktik yang berjudul "Rancang Bangun Aplikasi Pengelolaan Katalog Sparepart Sepeda Motor Berbasis Website pada Bengkel Lena Garage." Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah Kerja Praktik pada Program Studi Sistem Informasi, Fakultas Teknologi dan Informatika, Universitas Dinamika. Melalui kegiatan kerja praktik ini, penulis dapat mengaplikasikan pengetahuan yang telah dipelajari selama perkuliahan, khususnya dalam bidang pengembangan sistem informasi, perancangan *database*, implementasi aplikasi berbasis web, serta analisis kebutuhan pengguna. Selain itu, pelaksanaan kerja praktik ini memberikan pengalaman berharga kepada penulis dalam memahami proses bisnis yang berjalan di industri jasa otomotif, khususnya bengkel sepeda motor, serta bagaimana teknologi informasi dapat dimanfaatkan untuk meningkatkan efisiensi operasional, akurasi data, dan kualitas pelayanan kepada pelanggan. Atas terselesaikannya laporan Kerja Praktik ini, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan keluarga tercinta yang senantiasa memberikan doa, dukungan moral, dan motivasi yang tiada henti kepada penulis.
2. Ibu Tan Amelia, S.Kom., M.MT., Ph.D. selaku Dekan Fakultas Teknologi dan Informatika yang telah memberikan arahan dan fasilitas selama proses perkuliahan.





3. Ibu Endra Rahmawati, M.Kom, selaku Ketua Program Studi S1 Sistem Informasi yang telah memberikan bimbingan dan dukungan selama berlangsungnya mata kuliah Kerja Praktik.
4. Ibu Pradita Maulidya Effendi, M.Kom., selaku dosen wali yang telah memberikan bimbingan akademik, motivasi, dan arahan demi terselesaiannya laporan Kerja Praktik ini tepat waktu.
5. Ibu Sulistiowati, S.Si., M.M., selaku dosen pembimbing yang telah mendampingi, membimbing, dan memberikan masukan yang sangat berharga sepanjang proses penyusunan laporan ini.
6. Bapak/Ibu pemilik dan seluruh staff Bengkel Lena Garaged yang telah memberikan kesempatan, kepercayaan, dan dukungan penuh kepada penulis untuk melaksanakan kegiatan Kerja Praktik serta memberikan akses terhadap informasi dan data yang diperlukan.
7. Teman-teman mahasiswa Program Studi Sistem Informasi dan seluruh pihak yang telah memberikan bantuan, dukungan, dan kerja sama dalam proses penyelesaian laporan ini, baik secara langsung maupun tidak langsung.

Penulis berharap laporan Kerja Praktik ini dapat memberikan manfaat bagi pembaca, menjadi referensi untuk pengembangan sistem informasi sejenis, serta dapat membantu Bengkel Lena Garaged dalam meningkatkan kualitas layanan dan efisiensi pengelolaan *data sparepart*.

Surabaya, 05 Januari 2026

Penulis

DAFTAR ISI

Halaman

ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Manfaat	4
BAB II GAMBARAN UMUM PERUSAHAAN	5
2.1. Profil Perusahaan	5
2.2. Identitas Perusahaan.....	6
2.3. Visi Perusahaan.....	6
2.4. Misi Perusahaan.....	7
2.5. Struktur Organisasi	7
BAB III LANDASAN TEORI.....	9
3.1 <i>Website</i>	9
3.2 HTML (HyperText Markup Language)	9
3.3 PHP (Hypertext Preprocessor)	10
3.4 MySQL.....	10



3.5 XAMPP	11
3.6 Visual Studio Code.....	11
3.7 Hubungan Antar Teknologi.....	12
BAB IV DESKRIPSI PEKERJAAN	13
4.1 Gambaran Umum Sistem.....	13
4.1.1 Identifikasi Pengguna	13
4.1.2 Kebutuhan Fungsional.....	14
4.2 Design.....	15
4.2.1 Diagram Alur Proses	15
4.2.2. UseCase Diagram	17
4.2.3 <i>Activity</i> Diagram.....	18
4.2.4 Diagram Sequence	25
4.2.5 Class Diagram	33
4.3 Implementasi Aplikasi.....	36
4.3.1 Halaman Login.....	36
4.3.2 Halaman Utama.....	37
4.3.3 Halaman <i>Sparepart (Admin)</i>	39
4.3.4 Halaman Tambah <i>Sparepart</i>	41
4.3.5 Pencarian <i>Sparepart</i>	42
4.3.6 Halaman <i>Sparepart (User)</i>	44
4.3.7 Halaman Hitung CC.....	45
4.4 Pengujian Sistem	47
BAB V PENUTUP.....	48
5.1. Kesimpulan	48
5.2. Saran	48



DAFTAR PUSTAKA	49
LAMPIRAN	50



DAFTAR TABEL

	Halaman
Tabel 4. 1 Pengguna sistem	14
Tabel 4. 2 Kebutuhan Fungsional <i>Admin</i>	14
Tabel 4. 3 Kebutuhan Fungsional <i>User</i>	14
Tabel 4. 4 Pengujian Sistem	47



DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Logo Lena Garage.....	6
Gambar 2. 2 Struktur Organisasi.....	7
Gambar 4. 1 Alur Proses Manual	15
Gambar 4. 2 Alur Proses Sistem.....	16
Gambar 4. 3 Usecase Diagram.....	17
Gambar 4. 4 <i>Activity</i> Diagram halaman Login (Admin)	19
Gambar 4. 5 <i>Activity</i> Diagram halaman Login (User)	20
Gambar 4. 6 <i>Activity</i> Diagram Checkstock	21
Gambar 4. 7 <i>Activity</i> Diagram <i>Update</i>	22
Gambar 4. 8 <i>Activity</i> Diagram Hapus	23
Gambar 4. 9 <i>Activity</i> Diagram <i>Create</i>	24
Gambar 4. 10 Diagram <i>Sequence</i> halaman login (admin).....	26
Gambar 4. 11 Diagram <i>Sequence</i> halaman login (user)	27
Gambar 4. 12 Diagram <i>Sequence</i> checkstock.....	28
Gambar 4. 13 Diagram <i>Sequence update</i>	30
Gambar 4. 14 Diagram <i>Sequence</i> hapus	31
Gambar 4. 15 Diagram <i>sequence Create</i>	33
Gambar 4. 16 Class Diagram	35
Gambar 4. 17 Halaman Login.....	37
Gambar 4. 18 Halaman utama.....	38
Gambar 4. 19 Halaman utama.....	39
Gambar 4. 20 Halaman <i>sparepart</i> (admin)	40



UNIVERSITAS
Dinamika

Gambar 4. 21 Halaman Tambah <i>Sparepart</i>	42
Gambar 4. 22 Pencarian <i>Sparepart</i>	44
Gambar 4. 23 Halaman <i>sparepart</i> (user)	45
Gambar 4. 24 Halaman hitung cc	46

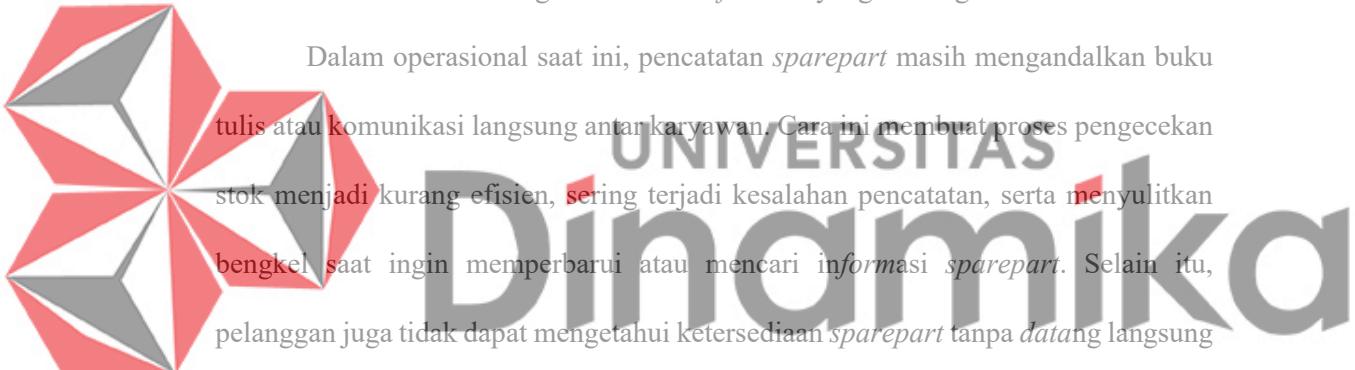


BAB I

PENDAHULUAN

1.1. Latar Belakang

Lena Garage adalah sebuah bengkel sepeda motor yang berlokasi di Dinoyo Baru No. 53 A, Surabaya. Sebagai bengkel yang melayani perbaikan dan penjualan *sparepart*, Lena Garage memiliki aktivitas yang cukup padat setiap harinya. Salah satu kegiatan penting di bengkel ini adalah mengelola *data sparepart*, mulai dari pendataan stok, pencarian barang, Namun, proses tersebut masih dilakukan secara manual dan belum didukung oleh sistem *informasi* yang terintegrasi.

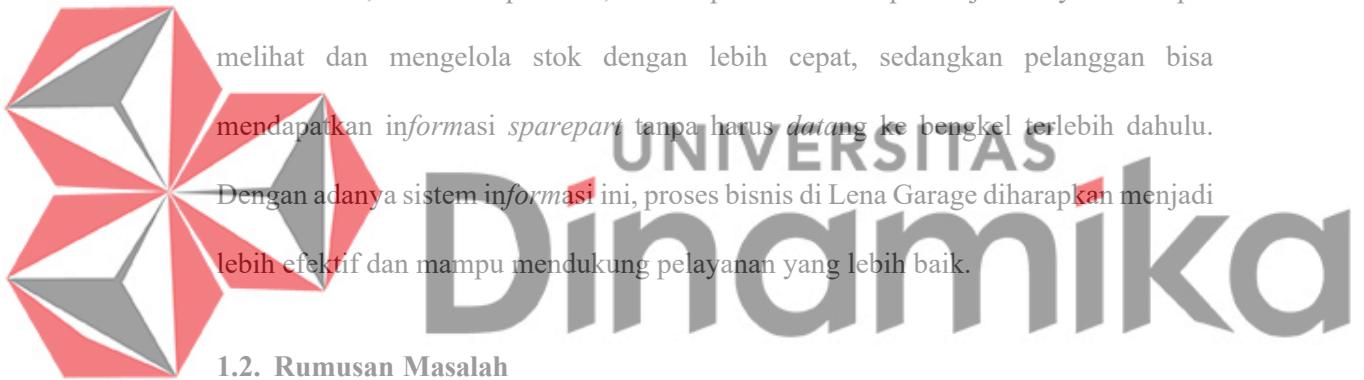


Dalam operasional saat ini, pencatatan *sparepart* masih mengandalkan buku tulis atau komunikasi langsung antarkaryawan. Cara ini membuat proses pengecekan stok menjadi kurang efisien, sering terjadi kesalahan pencatatan, serta menyulitkan bengkel saat ingin memperbarui atau mencari *informasi sparepart*. Selain itu, pelanggan juga tidak dapat mengetahui ketersediaan *sparepart* tanpa datang langsung ke bengkel, sehingga *informasi* yang diterima sering kali tidak cepat dan kurang praktis.

Berdasarkan kondisi tersebut, masalah utama yang muncul adalah belum adanya sistem yang dapat mengelola katalog *sparepart* secara rapi, cepat, dan mudah diakses. Ketiadaan sistem ini membuat proses kerja menjadi lebih lama dan peluang terjadinya salah *informasi* cukup besar. Bengkel juga kesulitan memberikan layanan yang lebih *informatif* kepada pelanggan karena keterbatasan dalam penyediaan *data* yang akurat.

Untuk mengatasi masalah tersebut, langkah yang digunakan adalah melakukan analisis kebutuhan di bengkel, memahami alur kerja yang berjalan, lalu merancang solusi berbasis sistem *informasi*. Metode yang digunakan meliputi observasi, pengumpulan *data*, perancangan sistem, serta pembuatan aplikasi sesuai kebutuhan bengkel. Pendekatan ini bertujuan agar sistem yang dibangun dapat menyelesaikan permasalahan yang ada dan sesuai dengan proses bisnis yang sebenarnya.

Solusi yang diusulkan adalah membuat aplikasi pengelolaan katalog *sparepart* berbasis *website*. Melalui sistem ini, *data sparepart* dapat dicatat dengan lebih teratur, mudah diperbarui, dan dapat diakses kapan saja. Karyawan dapat melihat dan mengelola stok dengan lebih cepat, sedangkan pelanggan bisa mendapatkan *informasi sparepart* tanpa harus *datang* ke bengkel terlebih dahulu. Dengan adanya sistem *informasi* ini, proses bisnis di Lena Garage diharapkan menjadi lebih efektif dan mampu mendukung pelayanan yang lebih baik.



1.2. Rumusan Masalah

1. Bagaimana cara LENA GARAGE menyediakan katalog *sparepart* yang bisa dilihat pelanggan tanpa harus *datang* langsung ke bengkel?
2. Bagaimana membuat sistem yang bisa menampilkan stok *sparepart* secara jelas dan selalu diperbarui?
3. Bagaimana menyajikan *informasi* jenis-jenis *sparepart*, harga, dan detail lainnya dalam bentuk *website* agar pelanggan lebih mudah mencari kebutuhan mereka?
4. Bagaimana membuat pengelolaan *data sparepart* di bengkel jadi lebih rapi dan tidak lagi bergantung pada pencatatan manual?

1.3. Batasan Masalah

1. Sistem yang dibuat hanya berfokus pada **pengelolaan katalog sparepart**, seperti nama barang, kategori, harga, dan ketersediaan stok.
2. *Website* tidak membahas proses **transaksi pembelian secara online**, melainkan hanya menampilkan *informasi sparepart* yang tersedia.
3. Pengelolaan *data sparepart* yang dilakukan dalam sistem hanya sebatas **input, edit, dan hapus data** oleh pihak bengkel.
4. Sistem tidak mencakup manajemen servis motor atau pencatatan pekerjaan mekanik, karena fokusnya hanya pada **data sparepart**.
5. *Informasi* yang ditampilkan pada *website* hanya berdasarkan **data yang diberikan pihak LENA GARAGE**, sehingga tidak mencakup *data* dari luar bengkel.
6. Aplikasi yang dibangun hanya dapat diakses melalui *web browser*, tanpa pengembangan versi mobile aplikasi (APK)



1.4. Tujuan

1. Memberikan kemudahan bagi LENA GARAGE dalam mengelola *data sparepart* agar lebih rapi dan mudah dicari ketika dibutuhkan.
2. Menyediakan sebuah katalog berbasis *website* yang bisa membantu pelanggan melihat jenis-jenis *sparepart* yang tersedia tanpa harus *datang* ke bengkel.
3. Membuat tampilan *informasi* stok yang lebih jelas dan teratur sehingga memudahkan pihak bengkel dalam memperbarui *data*.
4. Mendukung proses kerja di bengkel agar lebih efisien, terutama dalam hal pengecekan dan penyampaian *informasi sparepart*.

5. Membantu meningkatkan pelayanan kepada pelanggan dengan menyediakan akses *informasi* yang lebih cepat dan praktis.

1.5. Manfaat

1. Bagi Bengkel, pengelolaan *data sparepart* menjadi lebih teratur sehingga memudahkan dalam pengecekan dan pembaruan stok.
2. Bagi Pelanggan, mereka bisa melihat ketersediaan *sparepart* langsung dari *website* tanpa harus *datang* atau menanyakan dulu ke bengkel.
3. *Informasi sparepart* yang ditampilkan menjadi lebih jelas dan rapi, sehingga mengurangi kesalahpahaman terkait harga atau ketersediaan barang.
4. Proses kerja karyawan menjadi lebih cepat karena tidak perlu lagi mencari *data* secara manual dari catatan atau buku.

5. *Website* katalog bisa membantu meningkatkan profesionalitas dan kepercayaan pelanggan terhadap Lena Garage.

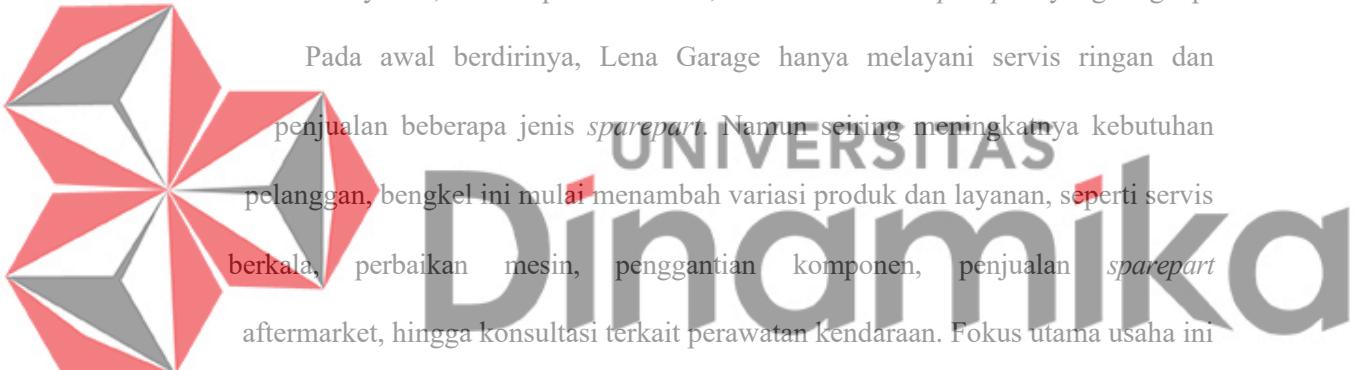


BAB II

GAMBARAN UMUM PERUSAHAAN

2.1. Profil Perusahaan

Lena Garage merupakan sebuah bengkel sepeda motor yang berlokasi di Dinoyo Baru No. 53 A, Surabaya. Bengkel ini didirikan dengan tujuan memberikan layanan perawatan, perbaikan, serta penyediaan *sparepart* untuk berbagai jenis sepeda motor. Seiring berjalannya waktu, LENA GARAGE berkembang menjadi salah satu bengkel yang cukup dikenal di wilayah tersebut karena mengedepankan kualitas layanan, keterampilan mekanik, dan ketersediaan *sparepart* yang lengkap.



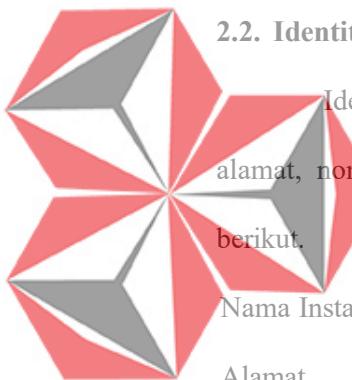
Pada awal berdirinya, Lena Garage hanya melayani servis ringan dan penjualan beberapa jenis *sparepart*. Namun seiring meningkatnya kebutuhan pelanggan, bengkel ini mulai menambah variasi produk dan layanan, seperti servis berkala, perbaikan mesin, penggantian komponen, penjualan *sparepart* aftermarket, hingga konsultasi terkait perawatan kendaraan. Fokus utama usaha ini tetap pada pelayanan servis motor dan penyediaan *sparepart* yang sesuai kebutuhan pelanggan.

Lena Garage juga menyediakan berbagai jenis *sparepart* seperti oli mesin, kampas rem, ban, *filter udara*, busi, dan komponen lainnya. Produk-produk tersebut ditampilkan pada katalog fisik di bengkel. Namun hingga saat ini, Lena Garage belum memiliki katalog digital sehingga pelanggan belum bisa melihat stok *sparepart* secara online.



Gambar 2. 1 Logo Lena Garage

Logo resmi dari Lena Garage dapat dilihat pada Gambar 2.1. Logo ini menjadi identitas visual bengkel yang digunakan pada papan nama, media sosial, dan dokumen internal



2.2. Identitas Perusahaan

Identitas perusahaan yang diinformasikan meliputi nama perusahaan, alamat, nomor telepon, serta email. Adapun rincian identitasnya adalah sebagai berikut.

Nama Instansi

: Megantara Narendra Parameswara

Alamat

: Dinoyo Baru no.53 A, Surabaya, Indonesia 60265

Nomor telepon

: 0813-3635-4456

Email

: parameswaramegantara@gmail.com

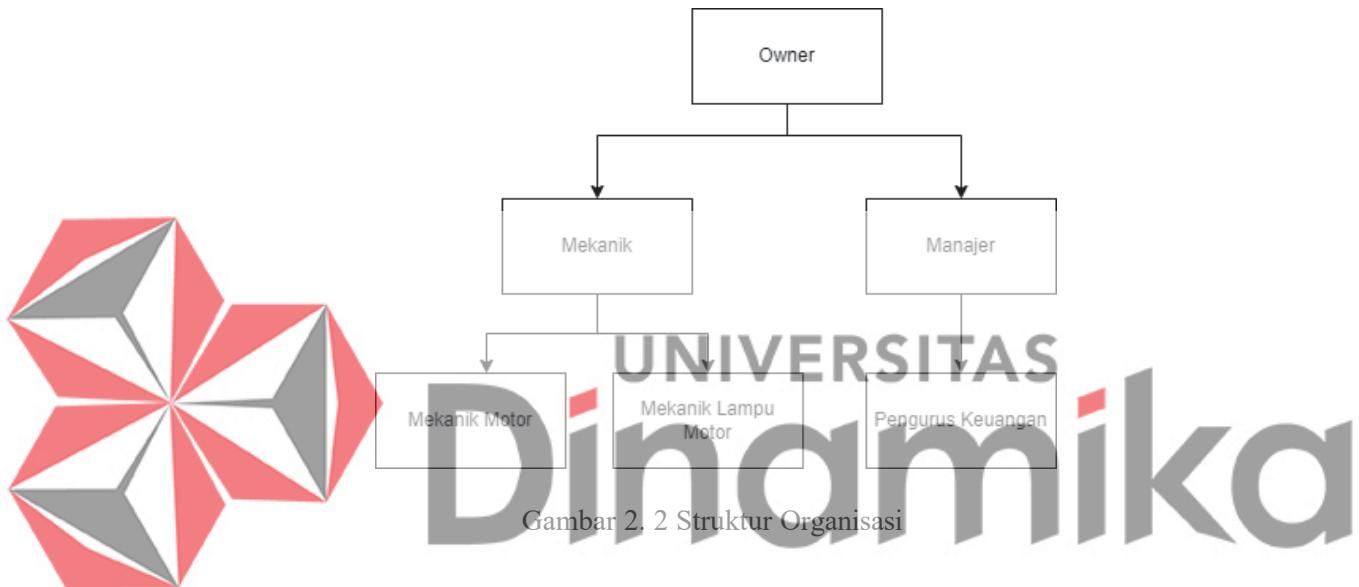
2.3. Visi Perusahaan

Visi perusahaan merupakan gambaran tujuan jangka panjang yang ingin dicapai oleh Lena Garage sebagai usaha di bidang jasa perawatan dan perbaikan kendaraan. Visi ini menjadi arah utama dalam setiap kegiatan operasional perusahaan serta sebagai pedoman dalam pengambilan keputusan dan pengembangan usaha di masa depan.

2.4. Misi Perusahaan

Misi perusahaan merupakan langkah-langkah strategis yang dilakukan oleh Lena Garage untuk mewujudkan visi yang telah ditetapkan. Misi ini mencerminkan komitmen perusahaan dalam memberikan pelayanan terbaik serta menjaga kepuasan dan kepercayaan pelanggan.

2.5. Struktur Organisasi



Struktur organisasi tersebut dipimpin oleh Owner sebagai pemilik usaha yang memiliki tanggung jawab tertinggi dalam menentukan arah, kebijakan, dan pengambilan keputusan strategis. Owner membawahi dua bagian utama, yaitu Manajer dan Mekanik, yang masing-masing memiliki peran penting dalam menjalankan operasional usaha. Manajer bertugas mengelola kegiatan operasional harian serta mengawasi aspek *administrasi* dan keuangan, sedangkan Mekanik bertanggung jawab pada bidang teknis dan kualitas pekerjaan perbengkelan. Di bawah Manajer terdapat Pengurus Keuangan yang bertugas mengelola pemasukan, pengeluaran, serta membuat laporan keuangan. Sementara itu, di bawah Mekanik

terdapat Mekanik Motor dan Mekanik Lampu Motor yang memiliki tugas khusus dalam menangani servis mesin, perawatan motor, serta perbaikan sistem kelistrikan dan lampu. Struktur ini menunjukkan pembagian tugas yang jelas antara manajemen dan teknis agar operasional usaha berjalan efektif dan terkontrol.

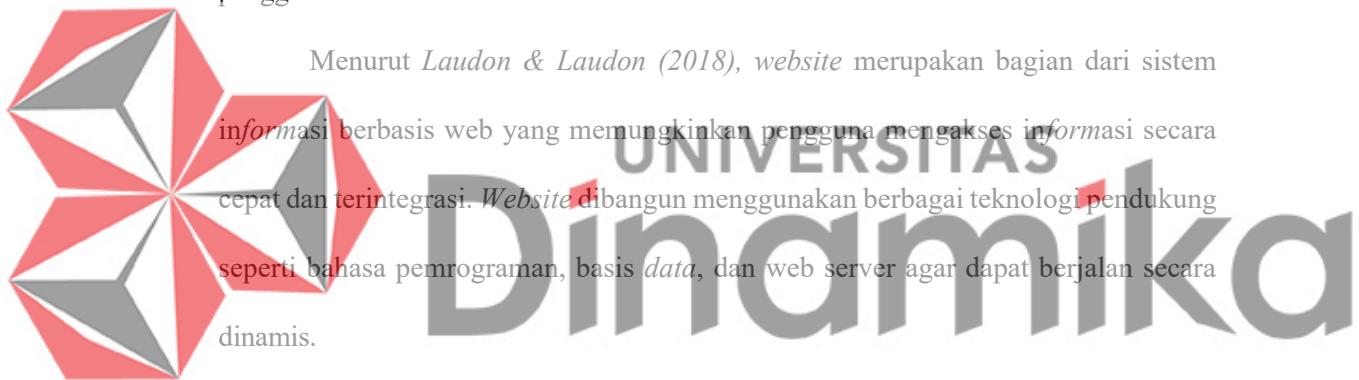


BAB III

LANDASAN TEORI

3.1 *Website*

Website merupakan kumpulan halaman web yang saling terhubung dan dapat diakses melalui jaringan internet menggunakan browser. *Website* digunakan sebagai media penyampaian *informasi*, layanan, maupun interaksi antara pengguna dan sistem. Dalam dunia bisnis dan industri, *website* berperan penting sebagai sarana digital untuk meningkatkan efisiensi, aksesibilitas *data*, dan pelayanan kepada pengguna.



Menurut Laudon & Laudon (2018), *website* merupakan bagian dari sistem *informasi* berbasis web yang memungkinkan pengguna mengakses *informasi* secara cepat dan terintegrasi. *Website* dibangun menggunakan berbagai teknologi pendukung seperti bahasa pemrograman, basis *data*, dan web server agar dapat berjalan secara dinamis.

Pada pelaksanaan Kerja Praktik ini, *website* digunakan sebagai media utama untuk mengelola dan menampilkan *data* secara terstruktur serta mempermudah proses pengolahan *informasi*.

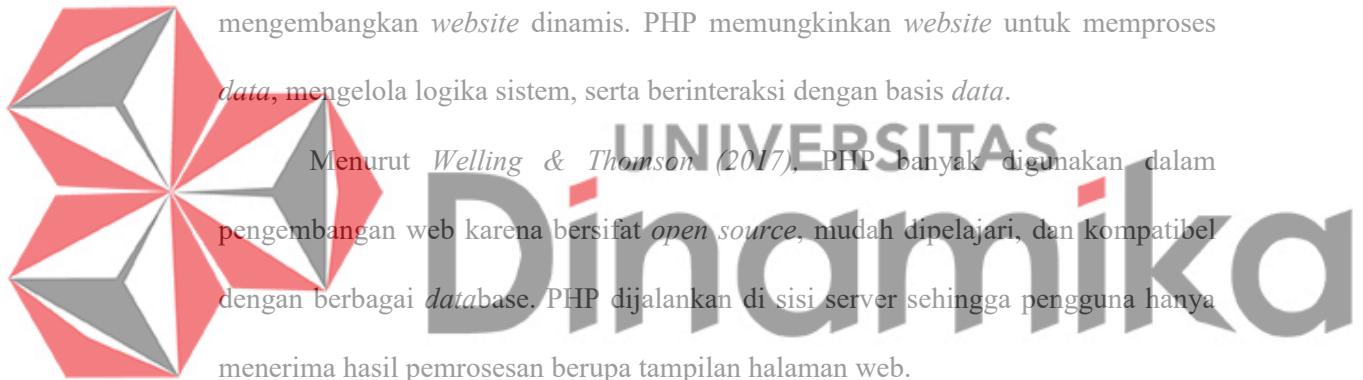
3.2 HTML (HyperText Markup Language)

HTML (HyperText Markup Language) merupakan bahasa markup standar yang digunakan untuk membuat struktur dasar halaman web. HTML berfungsi untuk menyusun elemen-elemen pada halaman web seperti teks, gambar, tabel, *form*, dan *hyperlink*.

HTML memungkinkan pengembang untuk mendefinisikan struktur dan konten sebuah halaman web agar dapat ditampilkan oleh browser dengan baik. HTML bekerja sama dengan teknologi lain seperti CSS dan JavaScript untuk menghasilkan tampilan web yang lebih menarik dan interaktif (Welling & Thomson, 2017). Dalam kerja praktik ini, HTML digunakan sebagai dasar pembuatan tampilan antarmuka (*user interface*) website agar informasi dapat disajikan secara jelas dan mudah dipahami oleh pengguna

3.3 PHP (Hypertext Preprocessor)

PHP merupakan bahasa pemrograman *server-side* yang digunakan untuk mengembangkan website dinamis. PHP memungkinkan website untuk memproses *data*, mengelola logika sistem, serta berinteraksi dengan basis *data*.



Menurut Welling & Thomson (2017), PHP banyak digunakan dalam pengembangan web karena bersifat *open source*, mudah dipelajari, dan kompatibel dengan berbagai *database*. PHP dijalankan di sisi server sehingga pengguna hanya menerima hasil pemrosesan berupa tampilan halaman web.

Dalam pelaksanaan Kerja Praktik, PHP digunakan untuk mengelola proses *input*, proses *data*, dan *output* pada website, seperti pengolahan *data* pengguna serta integrasi dengan *database* MySQL.

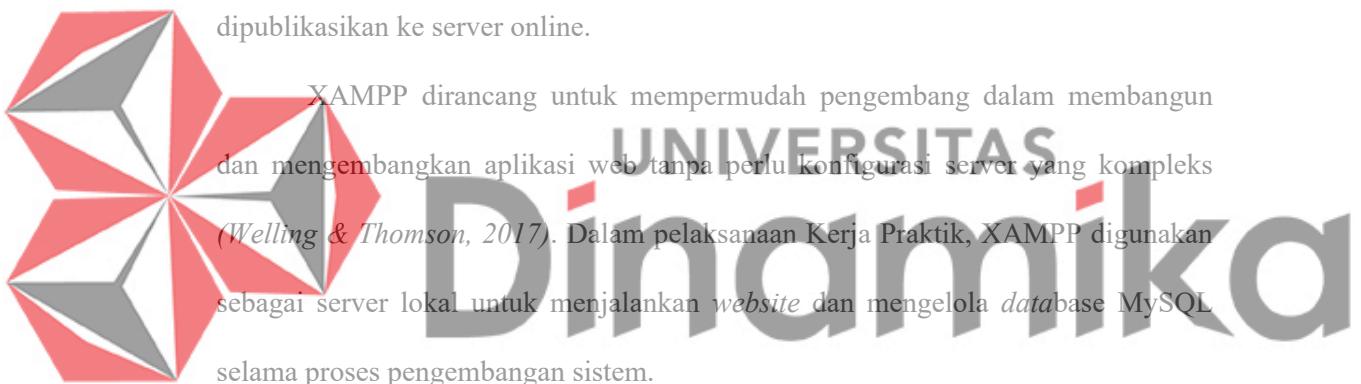
3.4 MySQL

MySQL merupakan sistem manajemen basis *data* relasional (RDBMS) yang digunakan untuk menyimpan dan mengelola *data* secara terstruktur. MySQL menggunakan bahasa SQL (*Structured Query Language*) untuk melakukan proses penyimpanan, pengambilan, pengubahan, dan penghapusan *data*.

Menurut *Silberschatz et al. (2019)*, database relasional memungkinkan *data* disimpan dalam tabel-tabel yang saling berhubungan sehingga mempermudah pengelolaan *data* secara efisien dan aman. Pada kerja praktik ini, MySQL digunakan sebagai media penyimpanan *data* utama yang diintegrasikan dengan PHP untuk mendukung pengolahan *data* pada *website*.

3.5 XAMPP

XAMPP merupakan paket perangkat lunak yang menyediakan lingkungan server lokal yang terdiri dari Apache, MySQL, PHP, dan Perl. XAMPP digunakan untuk menjalankan dan menguji *website* berbasis PHP secara lokal sebelum dipublikasikan ke server online.



3.6 Visual Studio Code

Visual Studio Code (VS Code) merupakan *text editor* yang digunakan untuk menulis dan mengelola kode program. VS Code mendukung berbagai bahasa pemrograman serta menyediakan fitur seperti *syntax highlighting*, *extensions*, dan *debugging tools* yang membantu pengembang meningkatkan produktivitas dalam pengembangan aplikasi karena tampilannya ringan dan fleksibel. Pada kerja praktik ini, Visual Studio Code digunakan sebagai alat utama dalam penulisan dan pengelolaan kode HTML, PHP, dan konfigurasi sistem *website*.

3.7 Hubungan Antar Teknologi

A. Integrasi HTML, PHP, dan MySQL

HTML berfungsi sebagai tampilan antarmuka, PHP sebagai pengolah logika sistem, dan MySQL sebagai penyimpan *data*. Ketiga teknologi ini saling terintegrasi untuk membentuk sebuah *website* dinamis yang mampu mengelola *data* secara efektif (*Laudon & Laudon, 2018*).

B. Peran XAMPP dan Visual Studio Code

XAMPP berperan sebagai server lokal untuk menjalankan *website*, sedangkan Visual Studio Code digunakan sebagai alat bantu pengembangan kode program.

Kombinasi kedua *tools* ini mendukung proses pengembangan *website* secara efisien selama Kerja Praktik.



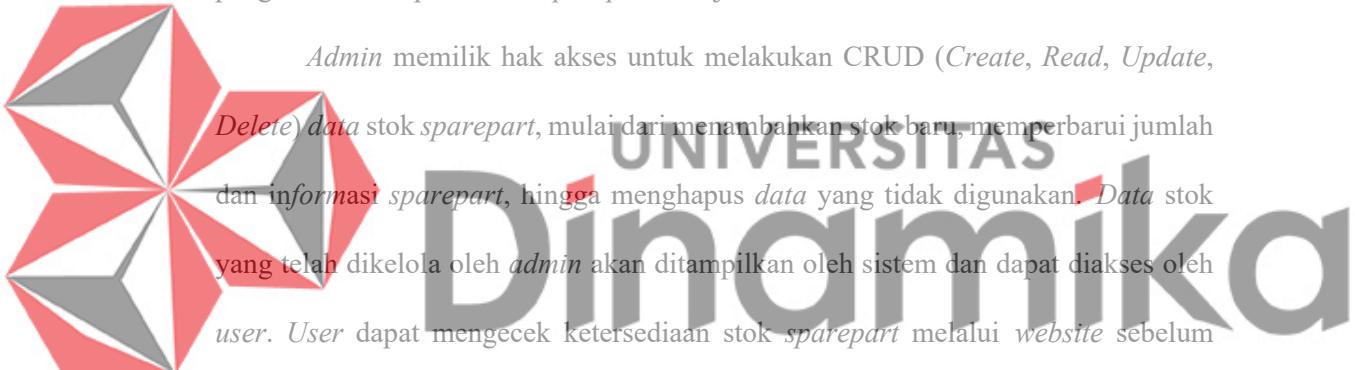
UNIVERSITAS
Dinamika

BAB IV

DESKRIPSI PEKERJAAN

4.1 Gambaran Umum Sistem

Sistem Web Bengkel Lena Garage merupakan sistem *informasi* berbasis web yang dirancang untuk mendukung pengelolaan stok *sparepart* dan memudahkan interaksi antara *admin* dan *user*. Sistem ini memiliki dua jenis pengguna, yaitu *admin* sebagai pengelola bengkel dan *user* sebagai pelanggan. Tujuan utama sistem adalah menyediakan *informasi* stok *sparepart* yang akurat dan *real-time* sehingga proses pengelolaan dan pembelian *sparepart* menjadi lebih efektif dan efisien.



Admin memiliki hak akses untuk melakukan CRUD (*Create, Read, Update, Delete*) *data* stok *sparepart*, mulai dari menambahkan stok baru, memperbarui jumlah dan *informasi* *sparepart*, hingga menghapus *data* yang tidak digunakan. *Data* stok yang telah dikelola oleh *admin* akan ditampilkan oleh sistem dan dapat diakses oleh *user*. *User* dapat mengecek ketersediaan stok *sparepart* melalui *website* sebelum datang ke bengkel untuk membeli, sehingga dapat menghemat waktu dan memastikan *sparepart* yang dibutuhkan tersedia. Sistem ini membantu meningkatkan pelayanan bengkel serta mempermudah pengelolaan stok dan *informasi* bagi semua pihak.

4.1.1 Identifikasi Pengguna

Sistem Pengelolaan Katalog Sparepart Lena Garage dirancang untuk melayani dua tipe pengguna dengan hak akses yang berbeda. Identifikasi pengguna dan kebutuhan fungsional masing-masing dapat dilihat pada tabel berikut :

Tabel 4. 1 Pengguna sistem

No	Tipe Pengguna	Deskripsi
1	<i>Admin</i>	Pengelola sistem yang dapat melakukan CRUD <i>data stock</i>
2	<i>User</i>	Pengguna yang hanya dapat melihat <i>data stock</i>

4.1.2 Kebutuhan Fungsional

A. Admin

Tabel 4. 2 Kebutuhan Fungsional *Admin*

No	Kebutuhan Fungsional	Deskripsi
1	<i>Login/Logout</i>	Admin dapat masuk dan keluar dari sistem
2	<i>Tambah Stock</i>	Admin dapat menambah <i>data stock</i> baru
3	<i>Edit Stock</i>	Admin dapat mengubah <i>data stock</i> yang ada
4	<i>Hapus Stock</i>	Admin dapat menghapus <i>data stock</i>
5	<i>Lihat Stock</i>	Admin dapat melihat daftar semua <i>stock</i>
6	<i>Search Stock</i>	Admin dapat mencari <i>data stock</i>
7	<i>Filter Stock</i>	Admin dapat memfilter <i>data stock</i> berdasarkan kategori tertentu
8	<i>Export Data Stock</i>	Admin dapat mengunduh <i>data stock</i>
9	<i>Kelola Profile</i>	Admin dapat mengubah <i>data profil</i> sendiri

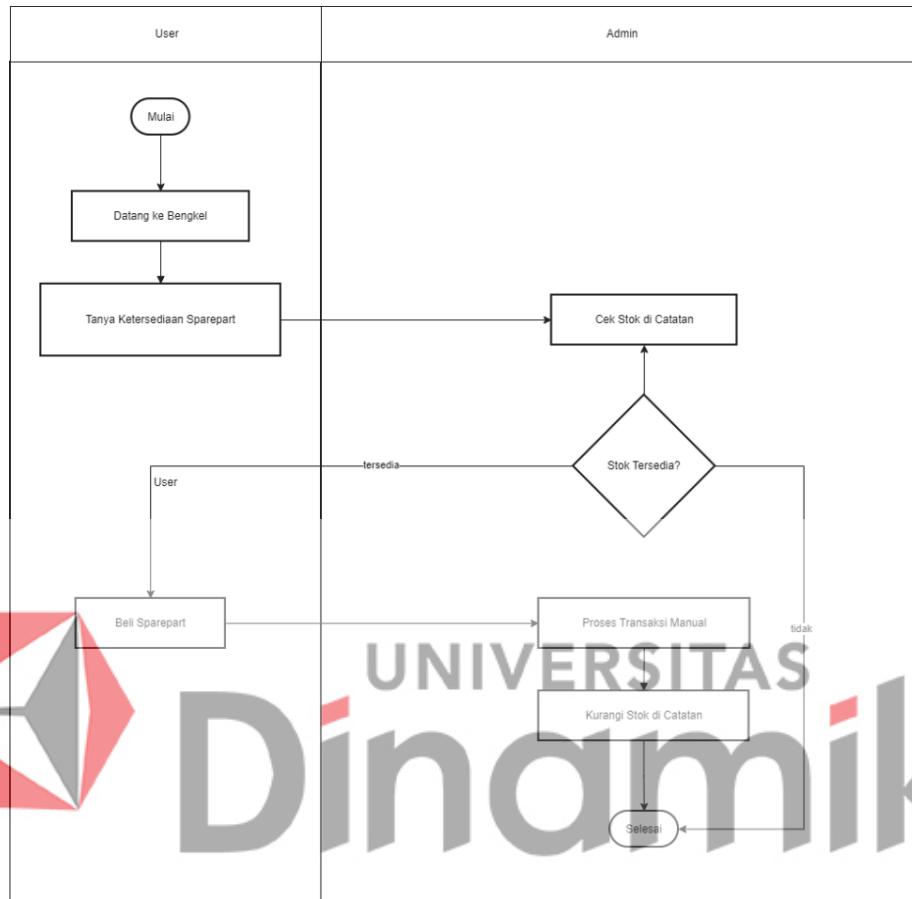
B. User

Tabel 4. 3 Kebutuhan Fungsional *User*

No	Kebutuhan Fungsional	Deskripsi
1	<i>Login/Logout</i>	User dapat masuk dan keluar dari sistem
2	<i>Lihat Stock</i>	User dapat melihat daftar semua <i>stock</i>
3	<i>Search Stock</i>	User dapat mencari <i>data stock</i>
4	<i>Filter Stock</i>	User dapat filter <i>data stock</i> berdasarkan kategori tertentu
5	<i>Lihat Detail Stock</i>	User dapat melihat detail informasi <i>stock</i>
6	<i>Kelola Profile</i>	User dapat mengubah <i>data profil</i> sendiri

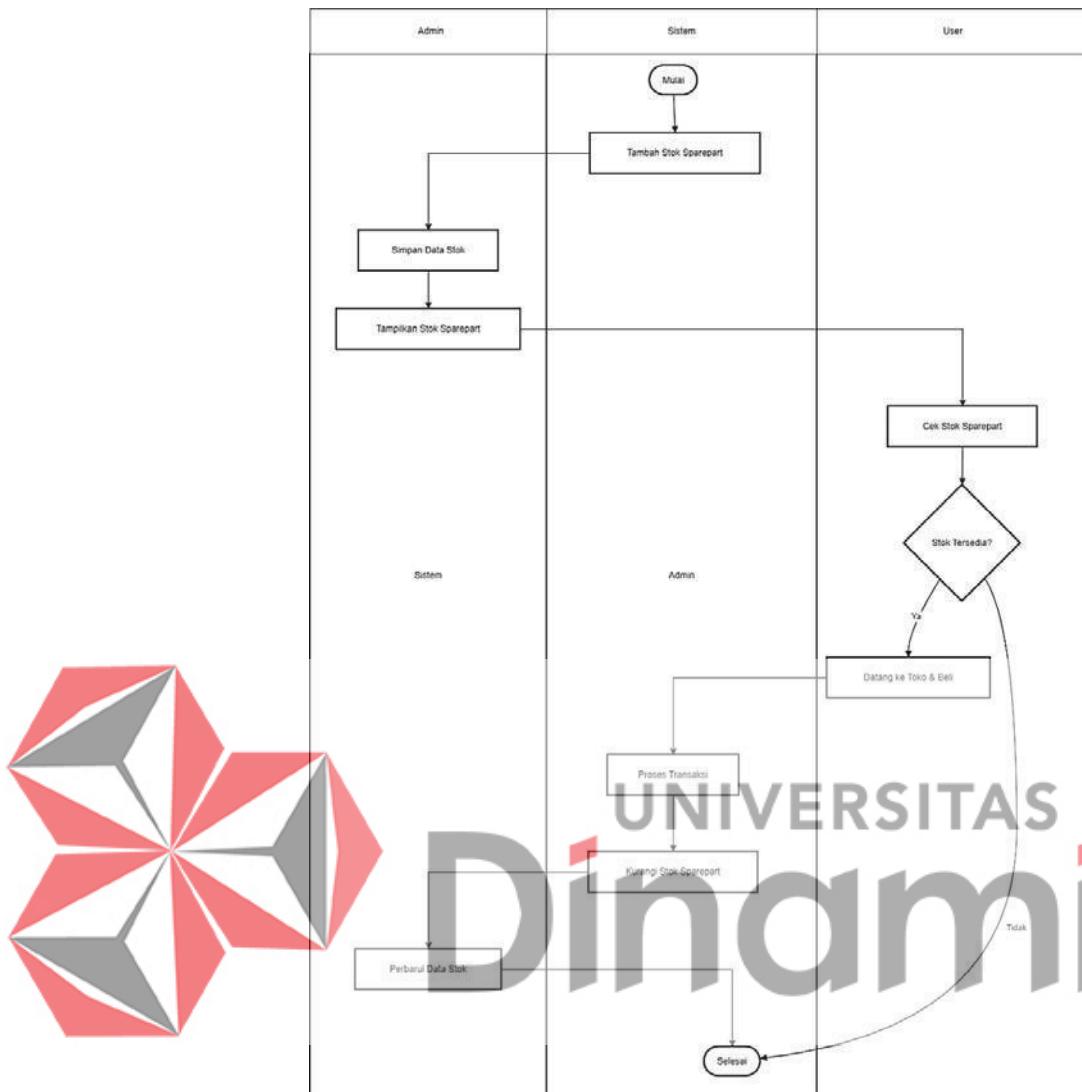
4.2 Design

4.2.1 Diagram Alur Proses



Gambar 4. 1 Alur Proses Manual

Pada Gambar 4.1 Alur bisnis manual dimulai saat *user* datang ke bengkel dan menanyakan ketersediaan *sparepart* kepada *admin*. *Admin* kemudian mengecek stok secara manual melalui catatan atau buku stok. Jika *sparepart* tersedia, *user* melakukan pembelian, *admin* memproses transaksi secara manual, dan stok dikurangi pada catatan. Jika stok tidak tersedia, proses berakhir tanpa transaksi.



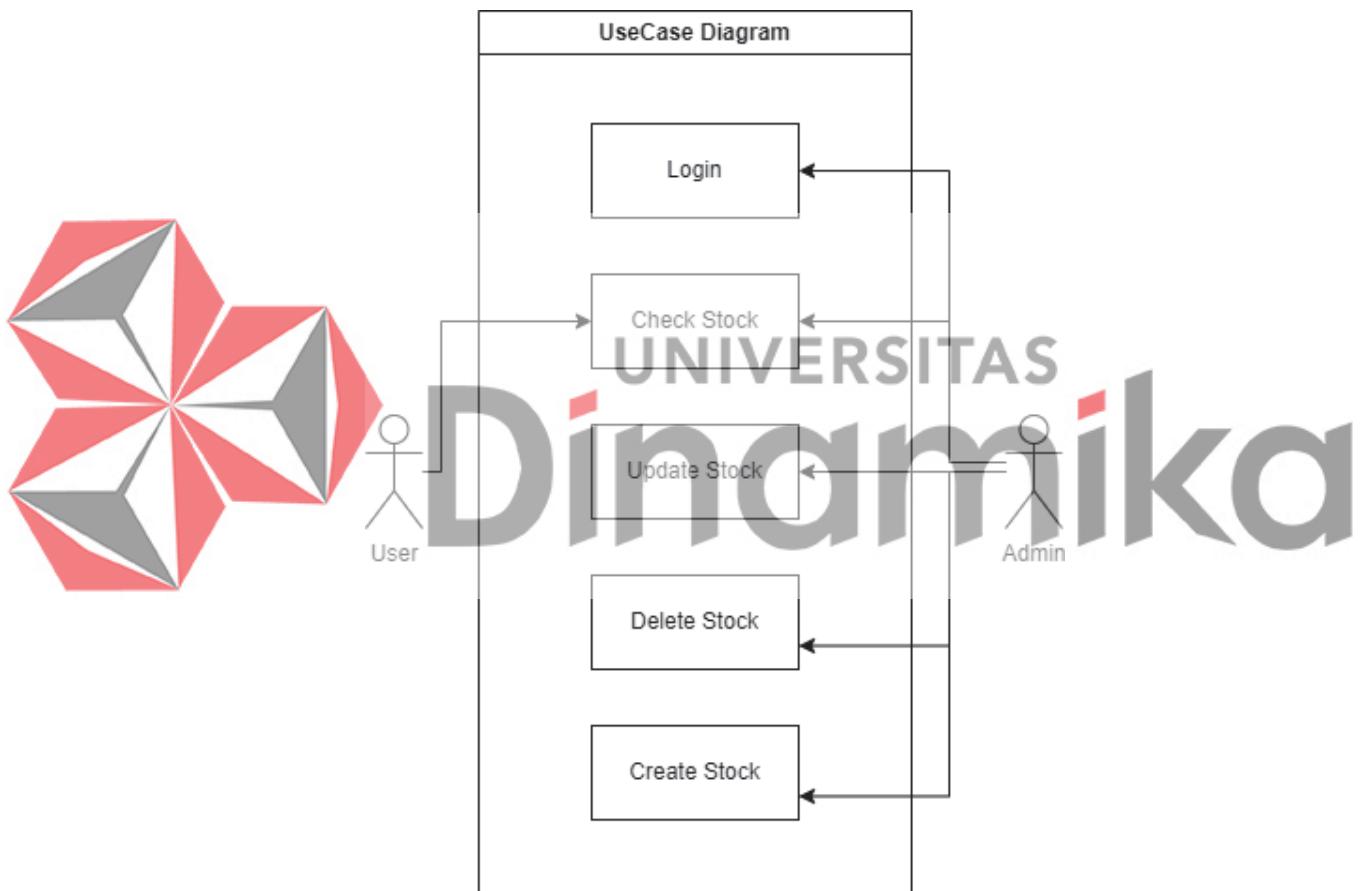
Gambar 4. 2 Alur Proses Sistem

Pada Gambar 4.2 Alur bisnis dimulai ketika *admin* bengkel menambahkan stok *sparepart* secara manual dengan cara mencatat barang yang masuk ke dalam buku stok atau catatan gudang. *Admin* juga melakukan pengecekan fisik *sparepart* untuk memastikan jumlah barang sesuai dengan catatan. *Data* stok yang tersedia hanya diketahui melalui catatan tersebut dan tidak ditampilkan secara online.

Selanjutnya, *user* yang ingin membeli *sparepart* harus *datang langsung* ke lokasi bengkel untuk menanyakan ketersediaan barang. *Admin* kemudian mengecek

stok secara manual berdasarkan catatan dan kondisi barang di gudang. Jika stok tersedia, *user* dapat langsung melakukan pembelian dan *admin* memproses transaksi secara manual. Setelah transaksi selesai, *admin* mengurangi jumlah stok pada catatan sesuai dengan barang yang terjual. Namun, jika stok tidak tersedia, transaksi tidak dapat dilakukan dan proses berakhir.

4.2.2. UseCase Diagram



Gambar 4. 3 Usecase Diagram

Pada Gambar 4.3 *Use Case Diagram* tersebut menggambarkan interaksi antara *User* dan *Admin* dengan sistem pengelolaan stok. Pada diagram ini terdapat dua aktor, yaitu *User* dan *Admin*, yang memiliki hak akses berbeda terhadap fitur

sistem. *User* hanya memiliki satu use case, yaitu *Check Stock*, yang berarti *user* dapat melihat atau mengecek ketersediaan stok tanpa dapat melakukan perubahan *data*. Sementara itu, *Admin* memiliki hak akses penuh terhadap sistem, dimulai dari *Login* untuk masuk ke sistem, kemudian dapat melakukan *Create Stock*, *Update Stock*, dan *Delete Stock* untuk mengelola *data* stok. *Admin* juga dapat melakukan *Check Stock* untuk melihat kondisi stok saat ini. Diagram ini menunjukkan pemisahan peran yang jelas, di mana *user* bersifat *read-only*, sedangkan *admin* bertanggung jawab penuh dalam pengelolaan *data* stok.

4.2.3 *Activity Diagram*

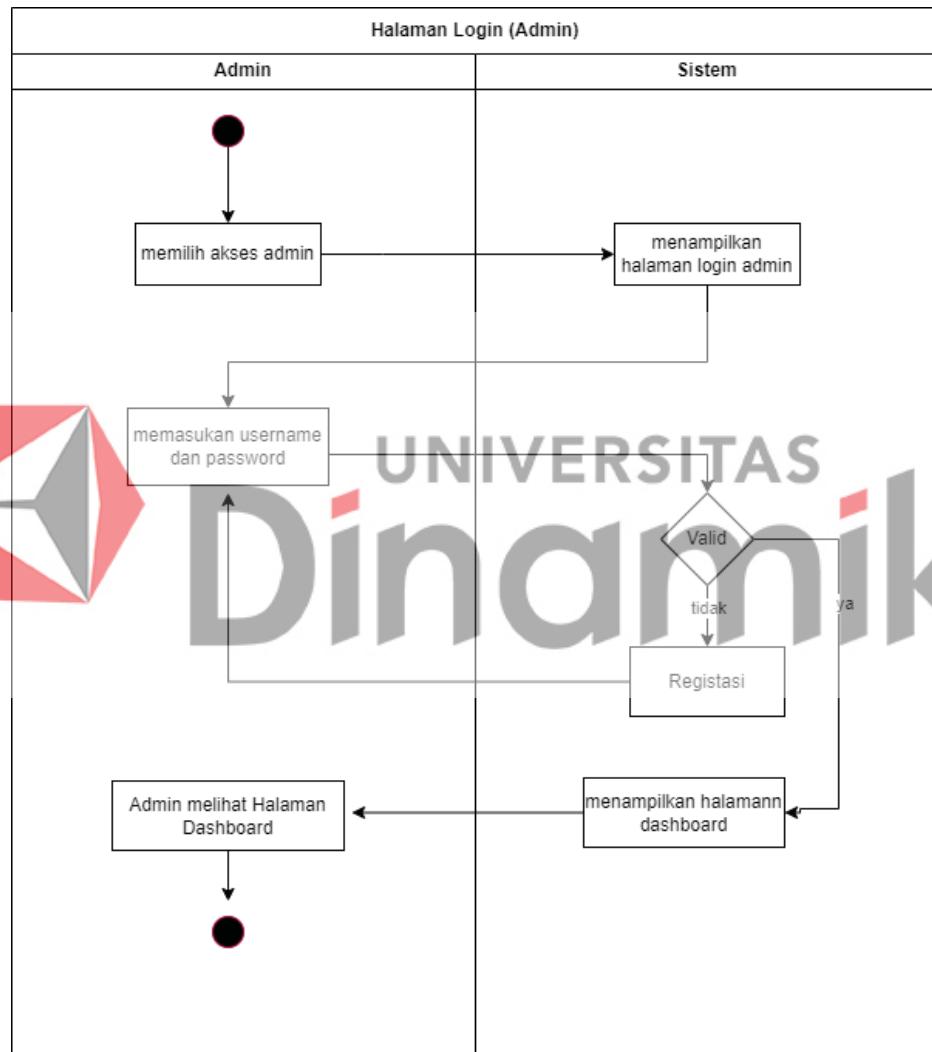


Activity diagram menggambarkan alur aktivitas atau proses kerja dalam sistem dari awal hingga akhir. Pada sistem pengelolaan stok bengkel, *activity* diagram menunjukkan langkah-langkah yang dilakukan oleh *admin* dan *user* secara berurutan serta keputusan yang mungkin terjadi selama proses tersebut. Proses dimulai ketika *admin* login ke sistem, kemudian *admin* dapat menambahkan, memperbarui, atau menghapus *data* stok sesuai kebutuhan. *Data* stok yang telah dikelola akan tersimpan dan siap ditampilkan. Sementara itu, *user* dapat melakukan pengecekan stok untuk mengetahui ketersediaan *sparepart* tanpa bisa mengubah *data*. *Activity* diagram ini membantu menjelaskan alur kerja sistem secara jelas, menunjukkan pembagian peran antara *admin* dan *user*, serta memastikan proses pengelolaan dan pengecekan stok berjalan secara terstruktur.

A. Halaman Login (*Admin*)

Proses dimulai saat *admin* memilih akses *admin*, kemudian sistem menampilkan halaman login *admin*. Selanjutnya *admin* memasukkan *username* dan

password yang akan divalidasi oleh sistem. Jika *data* login tidak valid, *admin* diarahkan ke proses registrasi atau diminta untuk mengisi ulang *data* login. Namun, jika *data* login valid, sistem akan menampilkan halaman Utama sehingga *admin* dapat mengakses dan melihat Utama, menandai berakhirnya alur proses login Pada gambar 4.4 ditunjukkan *activity diagram* proses login *Admin* ke dalam sistem.

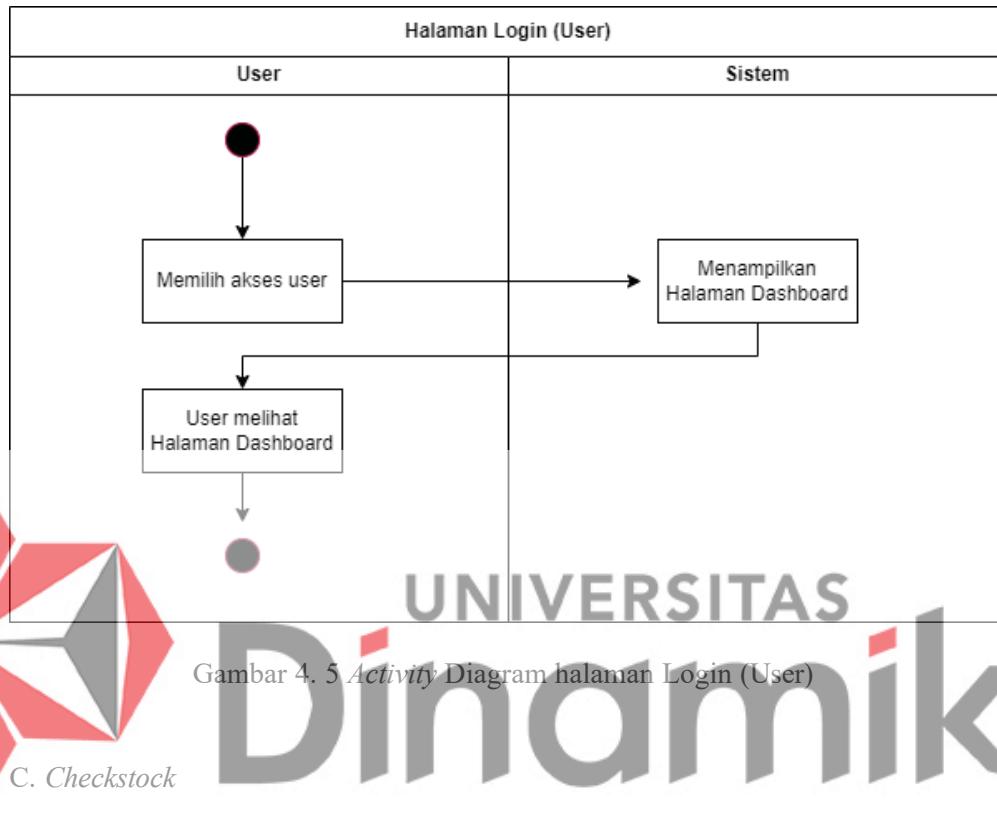


Gambar 4. 4 *Activity Diagram* halaman Login (Admin)

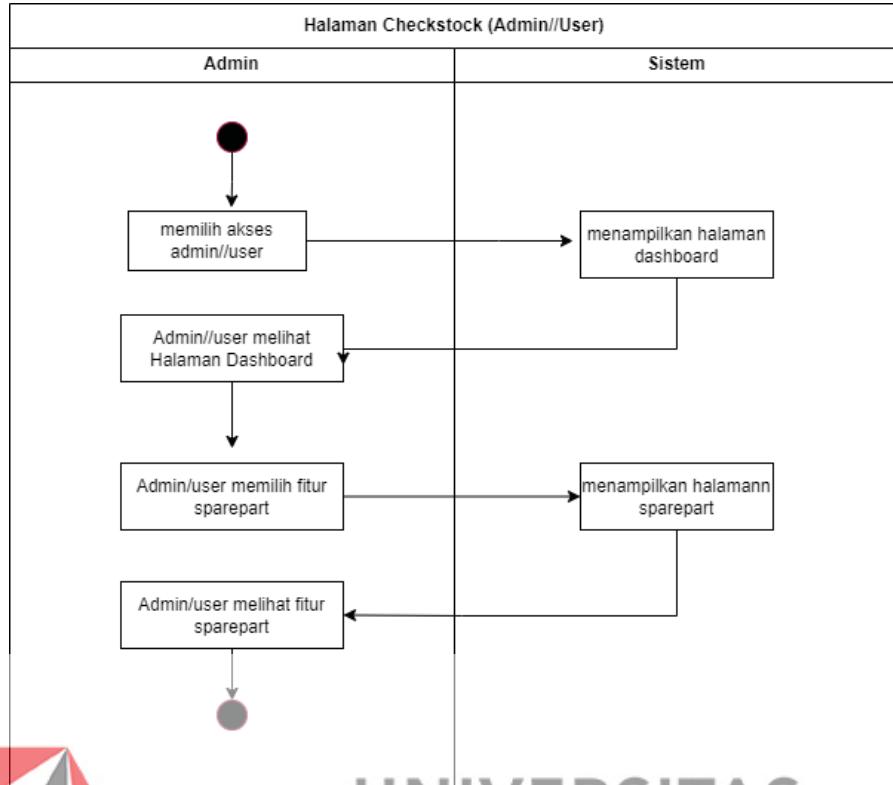
B. Halaman Login (*User*)

Proses dimulai ketika *user* memilih akses *user*, kemudian sistem langsung menampilkan halaman Utama. Setelah itu, *user* dapat melihat dan menggunakan

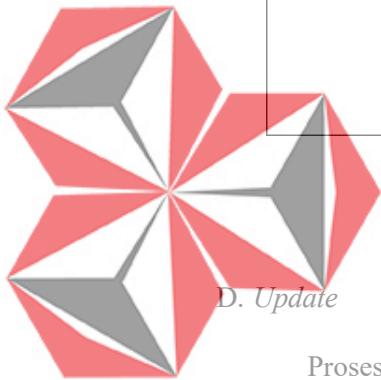
halaman Utama sesuai dengan hak akses yang dimiliki, dan proses berakhir ketika aktivitas *user* pada halaman tersebut selesai, Pada gambar 4.5 ditunjukkan *activity* diagram proses akses halaman *user* ke dalam sistem.



Proses dimulai ketika *admin* atau *user* memilih akses sesuai perannya, kemudian sistem menampilkan halaman Utama. Setelah melihat Utama, *admin* atau *user* memilih fitur *sparepart*, lalu sistem menampilkan halaman *sparepart*. Selanjutnya *admin* atau *user* dapat melihat informasi *sparepart* atau stok yang tersedia, dan proses berakhir setelah aktivitas pengecekan stok selesai Pada gambar 4.6 ditunjukkan *activity* diagram proses check stock yang dapat dilakukan oleh *admin* maupun *user*.

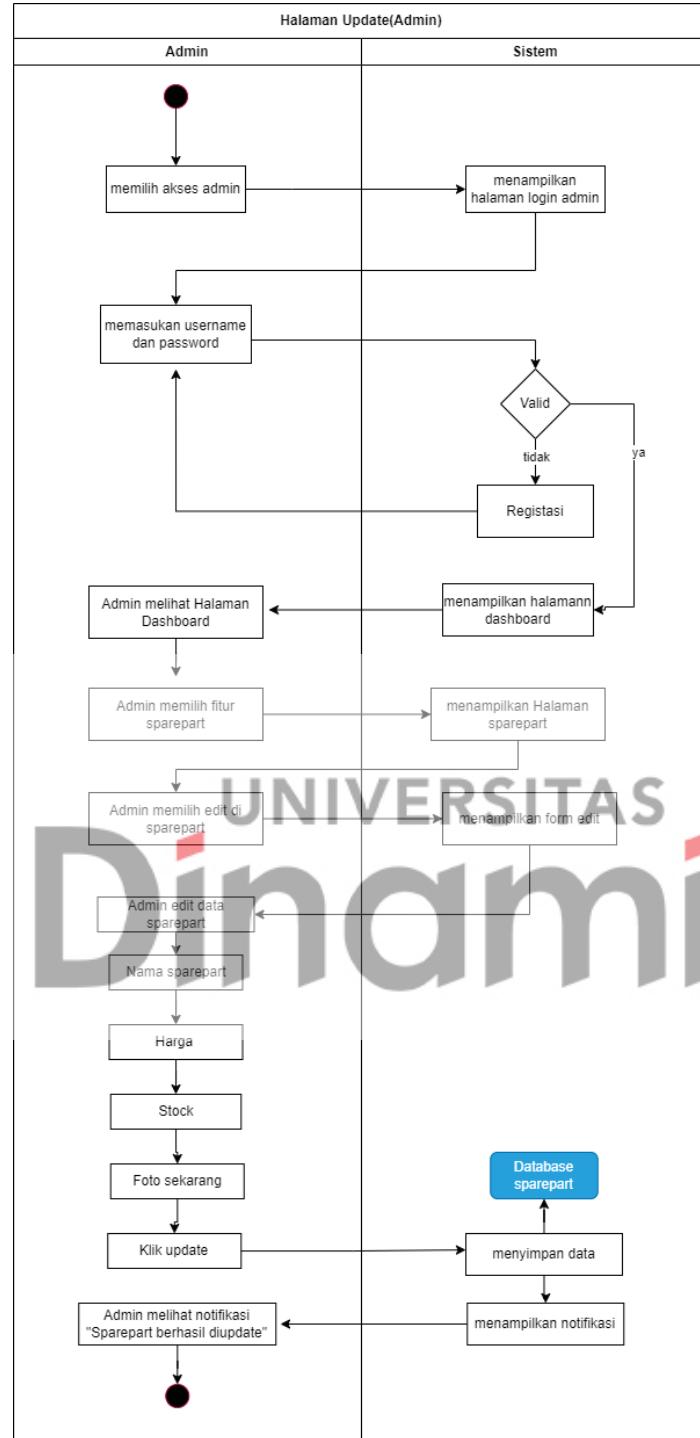
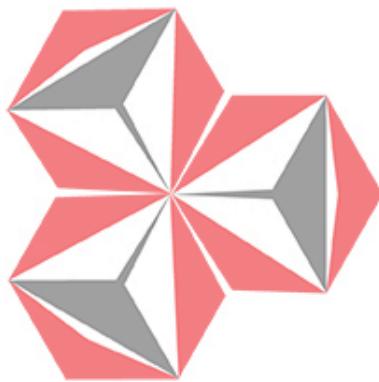


Gambar 4.6 Activity Diagram Checkstock



UNIVERSITAS
Dinamika

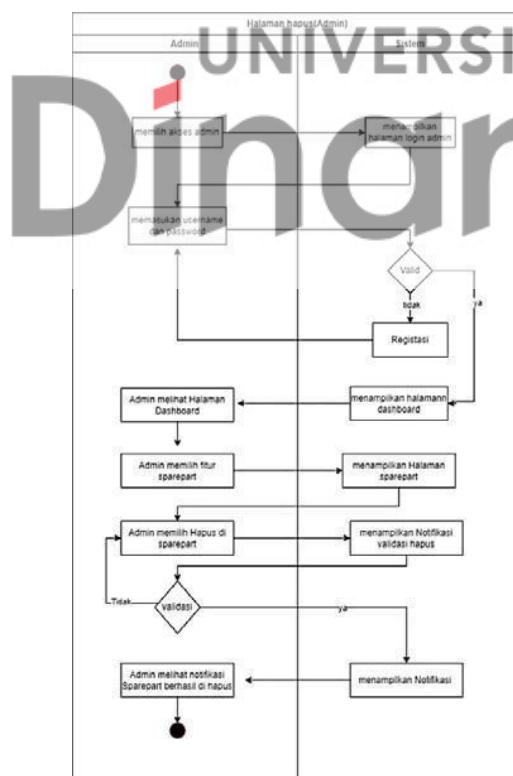
Proses dimulai ketika *admin* memilih akses *admin* dan melakukan *login* dengan memasukkan *username* serta *password*, kemudian sistem memvalidasi *data login*. Jika *data* valid, sistem menampilkan halaman Utama yang dapat diakses oleh *admin*. Selanjutnya *admin* memilih fitur *sparepart* dan memilih aksi *edit* pada *data sparepart*, sehingga sistem menampilkan *form edit*. *Admin* kemudian memperbarui *data* seperti nama *sparepart*, harga, stok, dan foto terbaru, lalu menekan tombol *update*. Sistem menyimpan perubahan ke dalam *database sparepart* dan menampilkan notifikasi bahwa *data sparepart* berhasil diupdate, yang menandai berakhirnya proses. Pada gambar 4.7 ditunjukkan *activity diagram* proses *update data sparepart* oleh *admin*.



Gambar 4. 7 Activity Diagram Update

E. Hapus

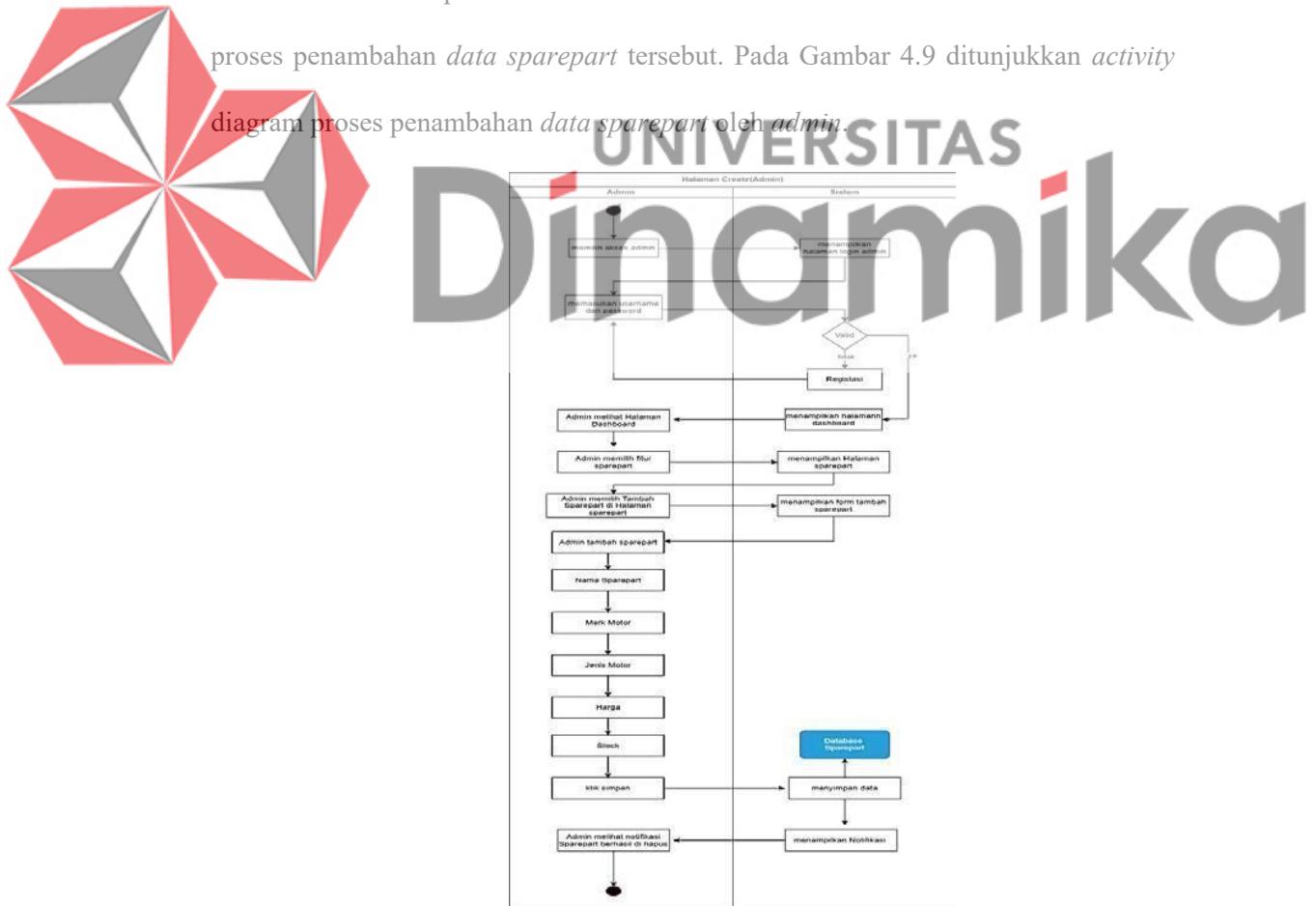
Proses dimulai ketika *admin* memilih akses *admin* dan melakukan login dengan memasukkan *username* serta password, kemudian sistem memvalidasi *data login*. Setelah berhasil masuk, sistem menampilkan halaman Utama dan *admin* memilih fitur *sparepart*. *Admin* kemudian memilih aksi hapus pada *data sparepart*, lalu sistem menampilkan notifikasi validasi penghapusan. Jika *admin* menyetujui proses validasi, sistem menampilkan notifikasi bahwa *data sparepart* berhasil dihapus. Namun, jika validasi ditolak, proses penghapusan dibatalkan. Alur proses berakhir setelah *admin* menerima notifikasi hasil dari tindakan penghapusan tersebut, Pada gambar 4.8 ditunjukkan *activity diagram* proses penghapusan *data sparepart* oleh *admin*.



Gambar 4. 8 *Activity Diagram Hapus*

F. Create

Proses dimulai ketika *admin* memilih akses *admin* dan melakukan login dengan memasukkan *username* serta *password*, kemudian sistem melakukan validasi *data login*. Setelah login berhasil, sistem menampilkan halaman Utama dan *admin* memilih fitur *sparepart*. Selanjutnya, *admin* memilih menu tambah *sparepart* sehingga sistem menampilkan *form* penambahan *data sparepart*. *Admin* kemudian mengisi *data sparepart* yang meliputi nama *sparepart*, merk motor, jenis motor, harga, dan stok, lalu menekan tombol simpan. Sistem menyimpan *data sparepart* ke dalam *database* dan menampilkan notifikasi bahwa *data sparepart* berhasil ditambahkan. Alur proses berakhir setelah *admin* menerima notifikasi hasil dari proses penambahan *data sparepart* tersebut. Pada Gambar 4.9 ditunjukkan *activity diagram* proses penambahan *data sparepart* oleh *admin*.



Gambar 4. 9 Activity Diagram *Create*

4.2.4 Diagram Sequence

Diagram *sequence* adalah salah satu jenis diagram *dalam Unified Modeling Language* (UML) yang digunakan untuk menggambarkan urutan interaksi antar aktor, objek, atau komponen sistem berdasarkan alur waktu. Diagram ini menampilkan bagaimana pesan atau perintah dikirim dari satu objek ke objek lain secara berurutan untuk menjalankan suatu proses atau skenario tertentu. Setiap objek digambarkan dengan *lifeline*, sedangkan komunikasi antar objek direpresentasikan dalam bentuk panah yang menunjukkan arah dan urutan pesan, sehingga aliran proses sistem dapat dipahami secara jelas dan terstruktur.

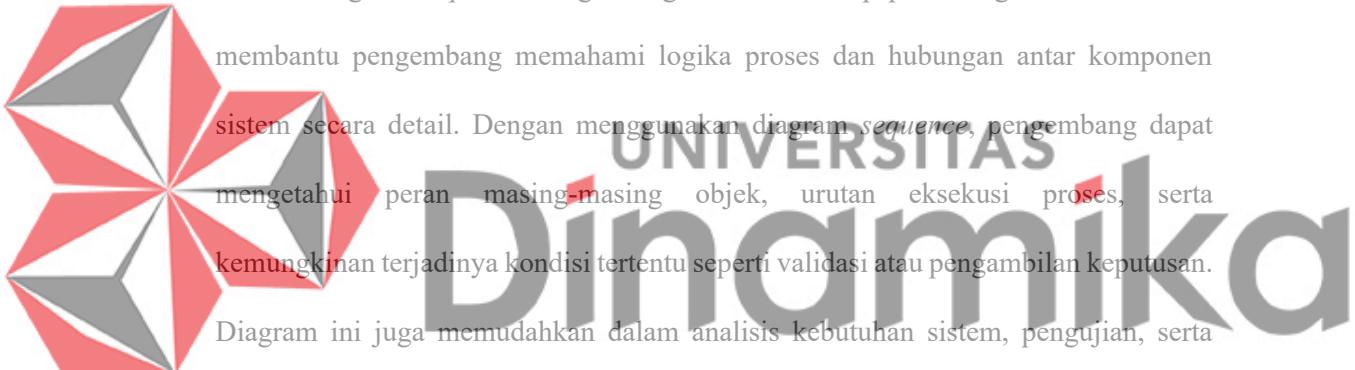
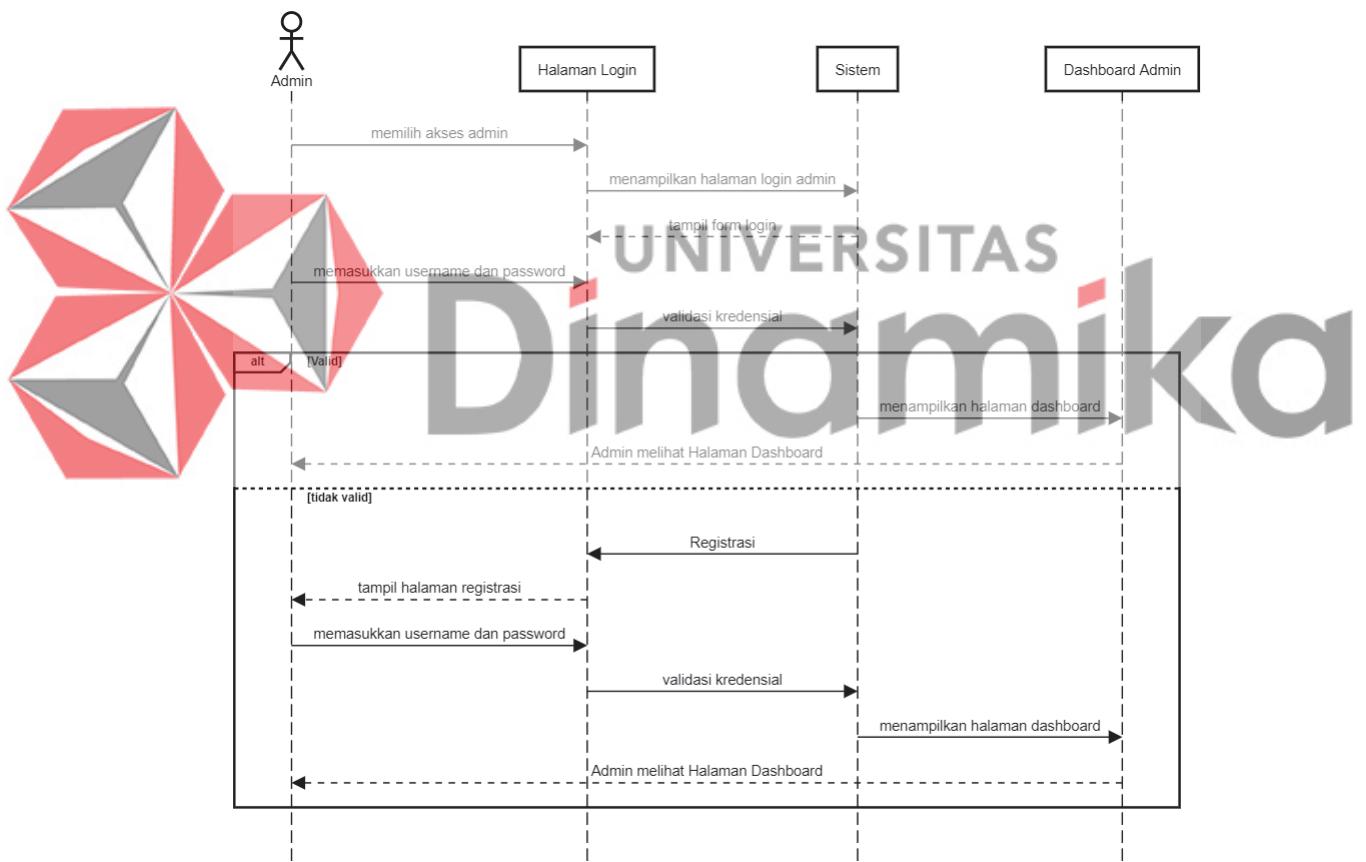


Diagram *sequence* sangat berguna dalam tahap perancangan sistem karena membantu pengembang memahami logika proses dan hubungan antar komponen sistem secara detail. Dengan menggunakan diagram *sequence*, pengembang dapat mengetahui peran masing-masing objek, urutan eksekusi proses, serta kemungkinan terjadinya kondisi tertentu seperti validasi atau pengambilan keputusan. Diagram ini juga memudahkan dalam analisis kebutuhan sistem, pengujian, serta dokumentasi, karena menggambarkan alur kerja sistem dari sudut pandang interaksi secara kronologis.

A. Halaman Login (*Admin*)

Proses dimulai ketika *admin* memilih akses *admin* dan sistem menampilkan halaman login *admin*, kemudian *admin* memasukkan *username* dan *password* untuk melakukan autentikasi. Sistem memvalidasi kredensial yang dimasukkan oleh *admin*, jika *data* login valid maka sistem langsung menampilkan halaman Utama *admin*, namun jika validasi tidak berhasil atau kredensial tidak valid, sistem mengarahkan *admin* ke halaman registrasi untuk mendaftarkan akun baru. Setelah

melakukan registrasi, *admin* kembali memasukkan *username* dan *password* yang baru dibuat, kemudian sistem melakukan validasi ulang dan menampilkan halaman Utama *admin*. Alur proses berakhir setelah *admin* berhasil melihat halaman Utama *admin* dan dapat mengakses seluruh fitur yang tersedia. Pada gambar di atas ditunjukkan *sequence diagram* proses login *admin* yang menggambarkan interaksi antara aktor *Admin* dengan sistem, mulai dari pemilihan akses *admin*, validasi kredensial, hingga tampilan *admin*, Pada Gambar 4.10 ditunjukkan Diagram *sequence* halaman login oleh *admin*.



Gambar 4. 10 Diagram *Sequence* halaman login (admin)

B. Halaman Login (*User*)

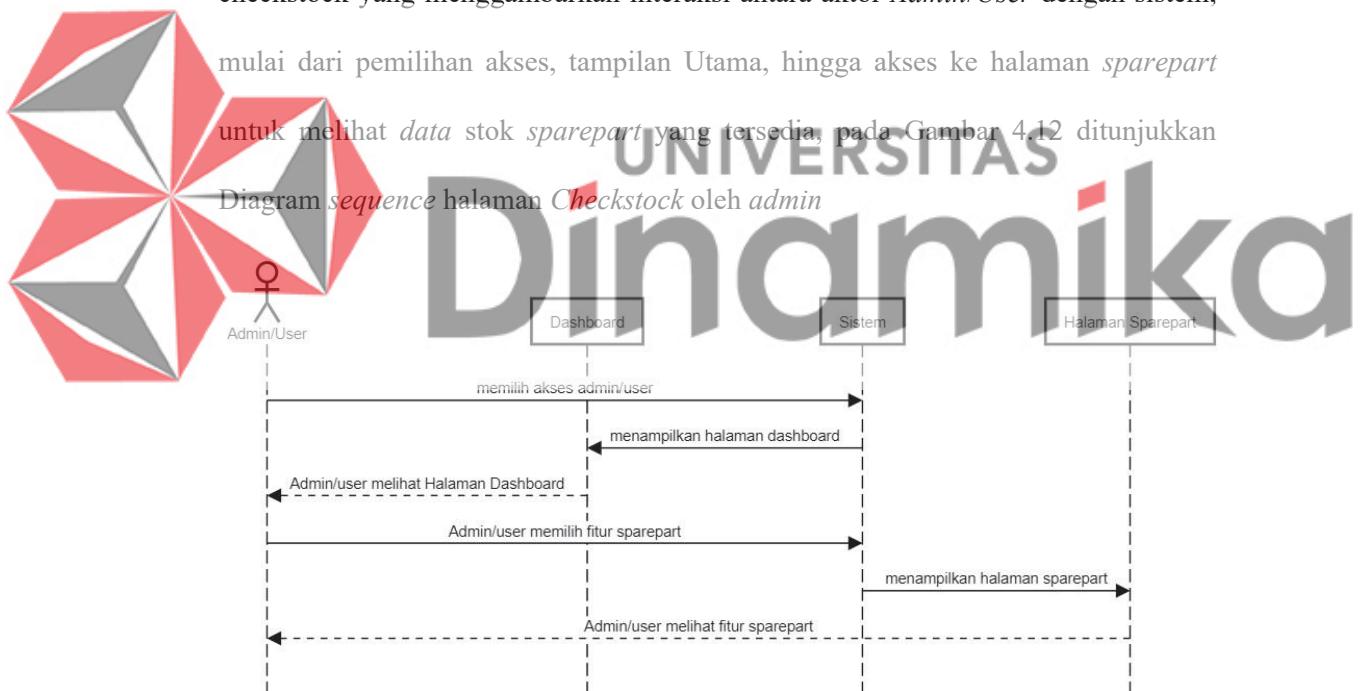
Proses dimulai ketika *user* memilih akses *user* untuk masuk ke dalam sistem, kemudian sistem langsung merespons dengan menampilkan halaman Tampilan *user* tanpa memerlukan proses login atau autentikasi. *User* kemudian dapat melihat halaman Utama yang berisi berbagai *informasi* dan fitur yang tersedia untuk *user*. Alur proses berakhir setelah *user* berhasil melihat dan mengakses halaman Utama *user*. Pada gambar di atas ditunjukkan *sequence* diagram proses akses *user* yang menggambarkan interaksi sederhana antara aktor *User* dengan sistem, mulai dari pemilihan akses *user* hingga tampilan halaman Utama *user*, pada Gambar 4.11 ditunjukkan Diagram *sequence* halaman login oleh *user*



Gambar 4. 11 Diagram *Sequence* halaman login (*user*)

C. Checkstock

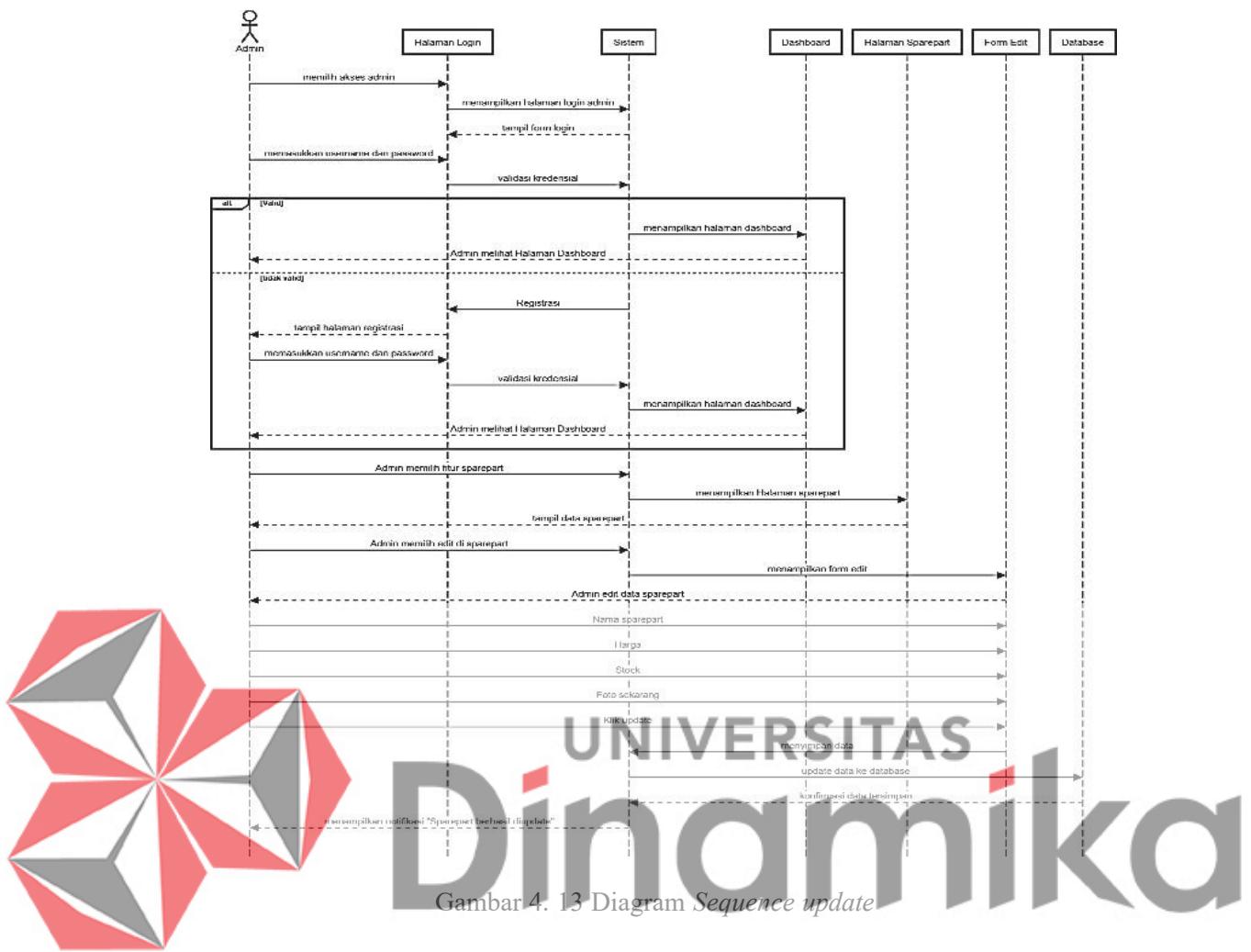
Proses dimulai ketika *admin* atau *user* memilih akses *admin/user* untuk masuk ke dalam sistem, kemudian sistem menampilkan halaman Utama dan *admin/user* dapat melihat halaman Utama yang berisi berbagai menu dan fitur yang tersedia. Setelah berhasil masuk ke Utama, *admin/user* memilih fitur *sparepart* untuk melihat *data sparepart* yang ada dalam sistem, kemudian sistem merespons dengan menampilkan halaman *sparepart* yang berisi daftar dan informasi lengkap mengenai *sparepart*. Alur proses berakhir setelah *admin/user* berhasil melihat dan mengakses halaman fitur *sparepart*. Pada gambar di atas ditunjukkan *sequence diagram* proses checkstock yang menggambarkan interaksi antara aktor *Admin/User* dengan sistem, mulai dari pemilihan akses, tampilan Utama, hingga akses ke halaman *sparepart* untuk melihat *data stok sparepart* yang tersedia, pada Gambar 4.12 ditunjukkan



Gambar 4. 12 Diagram Sequence checkstock

D. *Update*

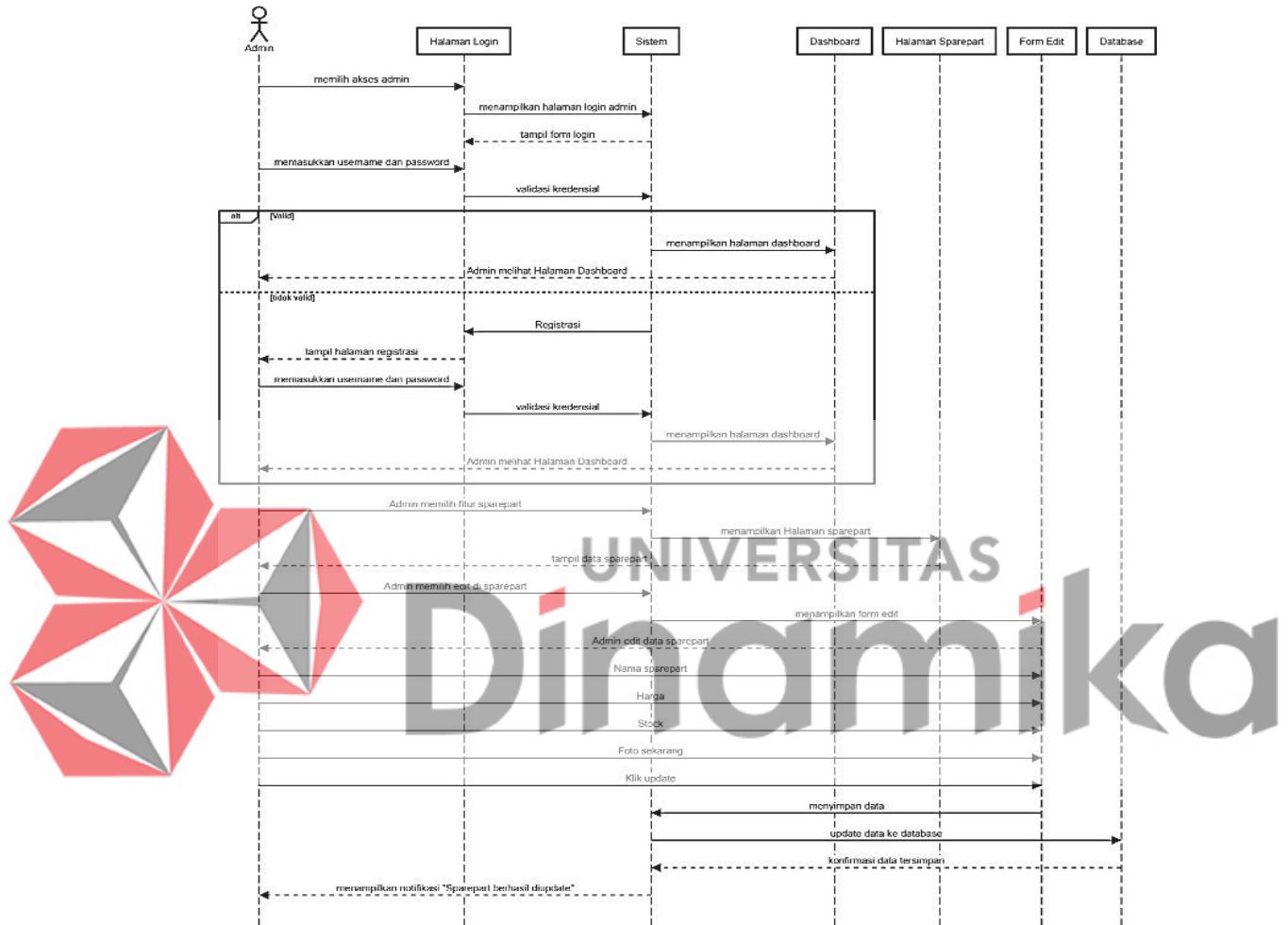
Proses dimulai ketika *admin* memilih akses *admin* dan melakukan login dengan memasukkan *username* serta *password*, kemudian sistem memvalidasi *data login*. Jika kredensial valid, sistem langsung menampilkan halaman Utama, namun jika tidak valid maka sistem mengarahkan *admin* ke halaman registrasi untuk mendaftarkan akun baru, setelah registrasi *admin*, sistem memasukkan *username* dan *password* lalu tidak menampilkan halaman Utama. Setelah berhasil masuk, *admin* memilih fitur *sparepart* dan akses menampilkan halaman *sparepart* yang berisi daftar *data sparepart*. *Admin* kemudian memilih aksi *edit* pada *data sparepart* yang ingin diubah, lalu tindak menampilkan *form edit* yang berisi *field data sparepart* seperti nama *sparepart*, harga, stock, dan foto sekarang. *Admin* melakukan perubahan *data* pada *form edit* tersebut dan mengklik tombol *update* untuk menyimpan perubahan, kemudian sistem menyimpan *data* yang telah diubah ke *database sparepart*. Setelah *data* berhasil tersimpan, tindak menampilkan notifikasi bahwa *sparepart* berhasil diupdate. Alur proses berakhir setelah *admin* menerima notifikasi hasil dari tindakan *update data sparepart* tersebut. Pada gambar di atas ditunjukkan *sequence diagram* proses *update data sparepart* oleh *admin* yang menggambarkan interaksi lengkap mulai dari login, akses Utama, pemilihan fitur *sparepart*, *edit data*, hingga penyimpanan *data* ke *database* pada Gambar 4.13 ditunjukkan Diagram *sequence* halaman login oleh *user*



E. Hapus

Proses dimulai ketika *admin* memilih akses *admin* dan melakukan login dengan memasukkan *username* serta *password*, kemudian sistem memvalidasi *data login*. Setelah berhasil masuk, sistem menampilkan halaman Utama dan *admin* memilih fitur *sparepart*. *Admin* kemudian memilih aksi *hapus* pada *data sparepart*, lalu sistem menampilkan notifikasi validasi penghapusan. Jika *admin* menyetujui proses validasi dengan memilih "Ya", sistem menghapus *data* dari *database* dan menampilkan notifikasi bahwa *sparepart* berhasil dihapus. Namun, jika validasi ditolak dengan memilih "Tidak", sistem mengembalikan *admin* ke halaman

sparepart tanpa menghapus *data*. Alur proses berakhir setelah *admin* menerima notifikasi hasil dari tindakan penghapusan tersebut. Pada gambar 4.14 ditunjukkan *sequence diagram* proses penghapusan *data sparepart* oleh *admin*.



Gambar 4. 14 Diagram *Sequence* hapus

F. Create

Proses dimulai ketika *admin* memilih akses *admin* dan melakukan login dengan memasukkan *username* serta *password*, kemudian sistem memvalidasi *data* login. Jika kredensial valid, sistem langsung menampilkan halaman Utama, namun jika tidak valid maka sistem mengarahkan *admin* ke halaman registrasi untuk mendaftarkan akun baru, setelah registrasi *admin* kembali memasukkan *username*

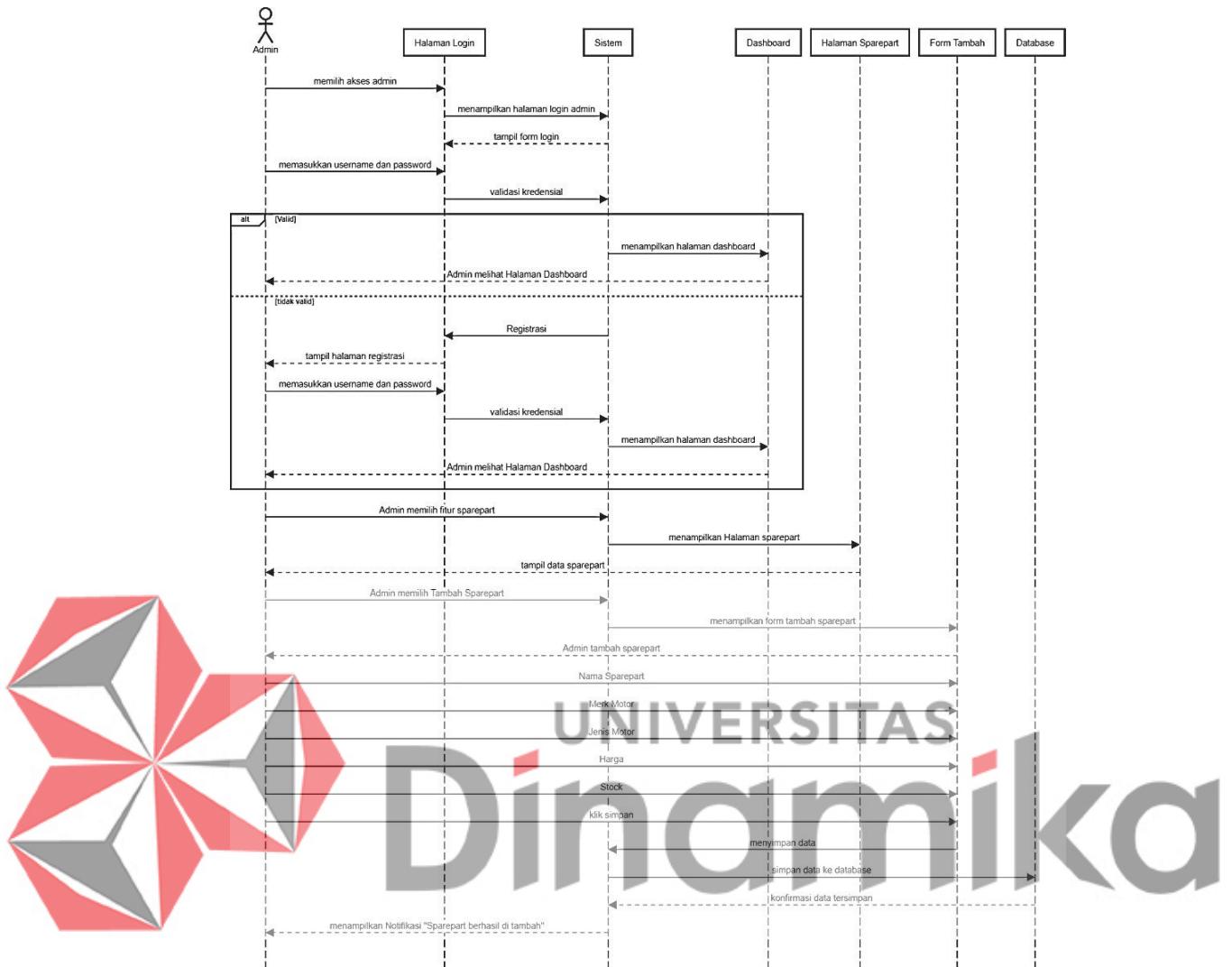
dan password lalu sistem menampilkan halaman Utama. Setelah berhasil masuk, *admin* memilih fitur *sparepart* dan sistem menampilkan halaman *sparepart* yang berisi daftar *data sparepart* yang tersedia.

Admin kemudian memilih tombol tambah *sparepart* di halaman *sparepart*, lalu sistem menampilkan *form* tambah *sparepart* yang berisi *field-field* input seperti nama *sparepart*, merk motor, jenis motor, harga, dan stock. *Admin* mengisi semua *data sparepart* yang diperlukan pada *form* tersebut dan mengklik tombol simpan untuk menyimpan *data* ke dalam sistem, kemudian sistem memproses *data* yang diinputkan dan menyimpannya ke *database* *sparepart*. Setelah *data* berhasil tersimpan di *database*, sistem menampilkan notifikasi bahwa *sparepart* berhasil ditambah. Alur proses berakhir setelah *admin* menerima notifikasi hasil dari tindakan penambahan *data sparepart* tersebut. Pada gambar di atas ditunjukkan *sequence diagram* proses penambahan *data sparepart* oleh *admin* yang menggambarkan interaksi lengkap mulai dari login hingga penyimpanan *data* ke *database*. Pada gambar 4.15 ditunjukkan *sequence diagram* proses penghapusan *data sparepart* oleh *admin*.



UNIVERSITAS

Dinamika



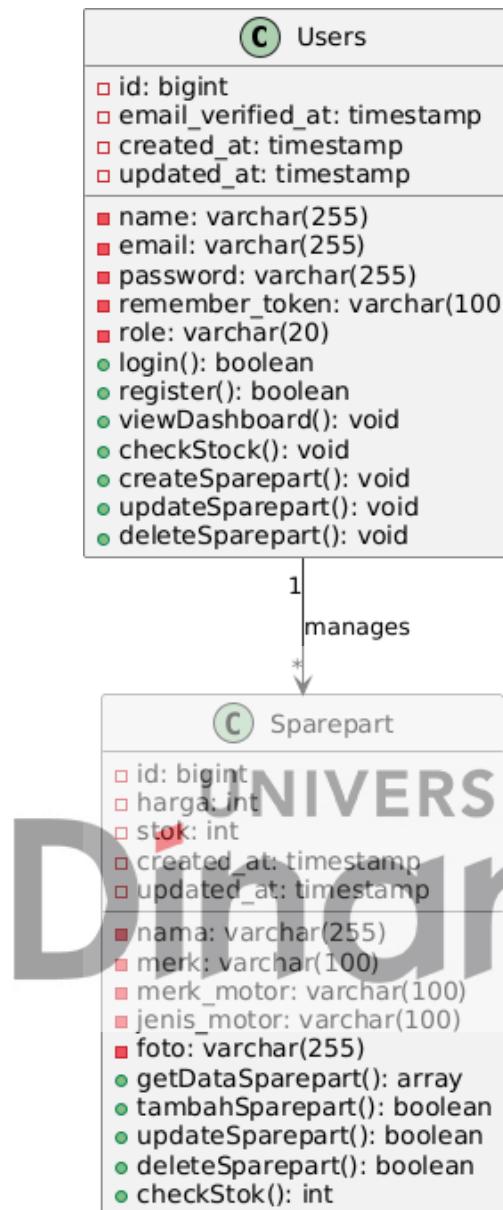
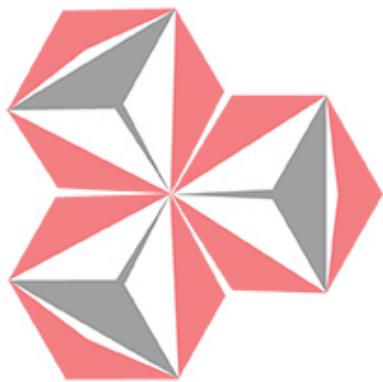
Gambar 4. 15 Diagram sequence Create

4.2.5 Class Diagram

Pada gambar 4.16 menggambarkan struktur dan hubungan antar class dalam sistem manajemen *sparepart* berdasarkan dua tabel *database* yang ada yaitu tabel *users* dan tabel *sparepart*. Class *Admin* merepresentasikan tabel *users* dengan atribut *id*, *name*, *email*, *email_verified_at*, *password*, *remember_token*, *Created_at*, *updated_at*, dan *role* dengan default value "user", serta memiliki method untuk melakukan login, *register*, melihat Utama, mengecek stock, dan mengelola *data*

sparepart (*Create*, *update*, *delete*). Class *Sparepart* merepresentasikan tabel *sparepart* dengan atribut *id*, *nama*, *merk*, *merk_motor*, *jenis_motor*, *harga*, *stok*, *foto*, *Created_at*, dan *updated_at*, dengan method untuk mengelola *data sparepart* seperti mengambil *data*, menambah, mengupdate, menghapus, dan mengecek stok *sparepart* yang tersedia. Hubungan antar class menunjukkan bahwa satu *Admin* dapat mengelola banyak *Sparepart* (relasi *one-to-many*). Pada gambar di atas ditunjukkan *class diagram* sistem manajemen *sparepart* yang merepresentasikan entitas, atribut, method, dan relasi antar class sesuai dengan struktur *database* yang telah dibuat.

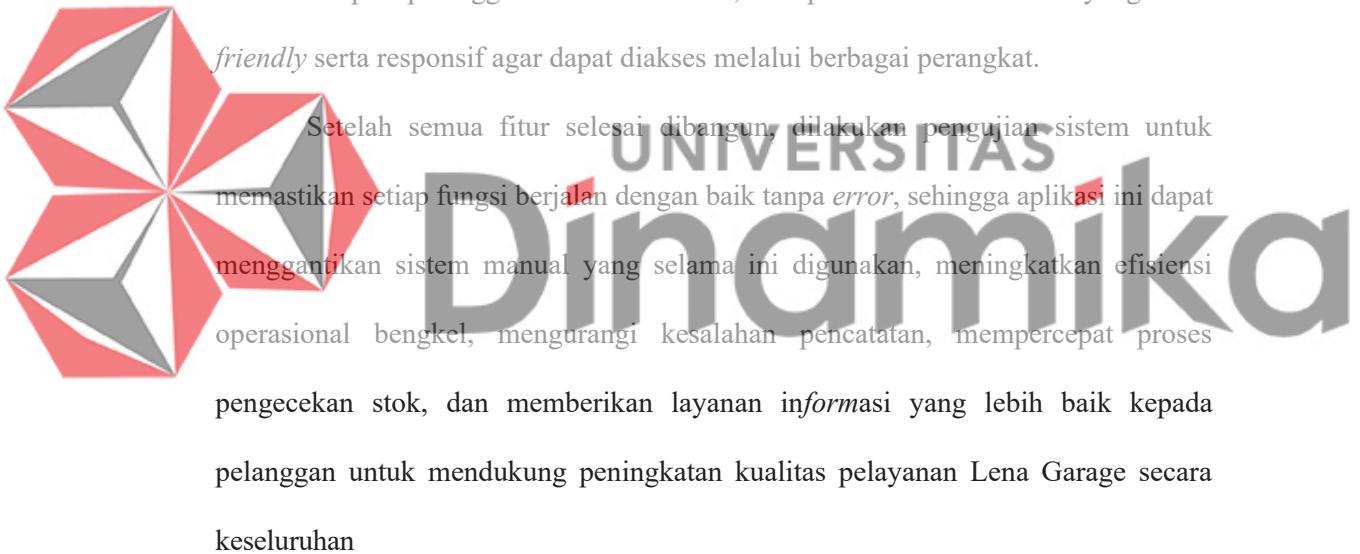




Gambar 4. 16 Class Diagram

4.3 Implementasi Aplikasi

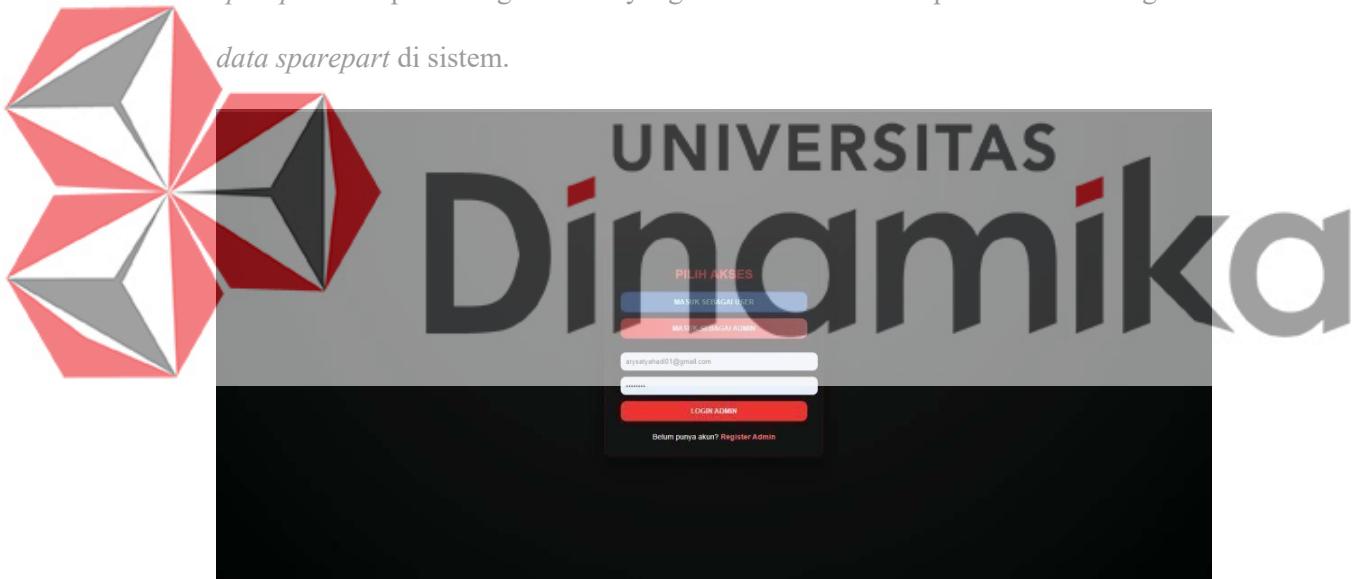
Implementasi aplikasi pengelolaan katalog *sparepart* di Lena Garage merupakan tahap realisasi dari perancangan sistem menjadi aplikasi berbasis *website* yang dibangun menggunakan teknologi HTML, CSS, JavaScript untuk tampilan antarmuka, PHP Laravel untuk logika pemrosesan *data*, dan MySQL sebagai *database*. Proses implementasi meliputi pembuatan struktur *database* dengan tabel *users* untuk *data admin* dan tabel *sparepart* untuk *informasi sparepart* lengkap, pengembangan fitur-fitur sistem seperti *login* dan *registrasi admin*, pengelolaan *sparepart* (tambah, ubah, hapus), serta pengecekan stok yang dapat diakses oleh *admin* maupun pelanggan secara *real-time*, dan pembuatan antarmuka yang *user-friendly* serta responsif agar dapat diakses melalui berbagai perangkat.



4.3.1 Halaman Login

Halaman ini merupakan halaman awal atau landing page dari aplikasi pengelolaan katalog *sparepart* Lena Garage berfungsi sebagai pintu masuk bagi pengguna untuk memilih jenis akses yang akan digunakan. Pada halaman ini terdapat dua pilihan akses utama yaitu tombol "MASUK SEBAGAI USER" dengan warna biru yang mengarahkan pengguna umum atau pelanggan untuk langsung

mengakses katalog *sparepart* tanpa perlu login, dan tombol "MASUK SEBAGAI *ADMIN*" dengan warna merah yang mengarahkan ke halaman login khusus *admin*. Ketika memilih akses *admin*, akan muncul *form* login yang terdiri dari *input field* untuk email dan password, tombol "LOGIN *ADMIN*" untuk melakukan autentikasi, serta link "Register *Admin*" di bagian bawah untuk *admin* baru yang belum memiliki akun dan perlu mendaftar terlebih dahulu. Halaman ini dirancang dengan tampilan yang sederhana dan jelas menggunakan background gelap (*dark theme*) agar pengguna dapat dengan mudah memahami pilihan akses yang tersedia dan memilih sesuai dengan kebutuhan mereka, baik sebagai *user* yang hanya ingin melihat katalog *sparepart* maupun sebagai *admin* yang memiliki hak akses penuh untuk mengelola *data sparepart* di sistem.



Gambar 4. 17 Halaman Login

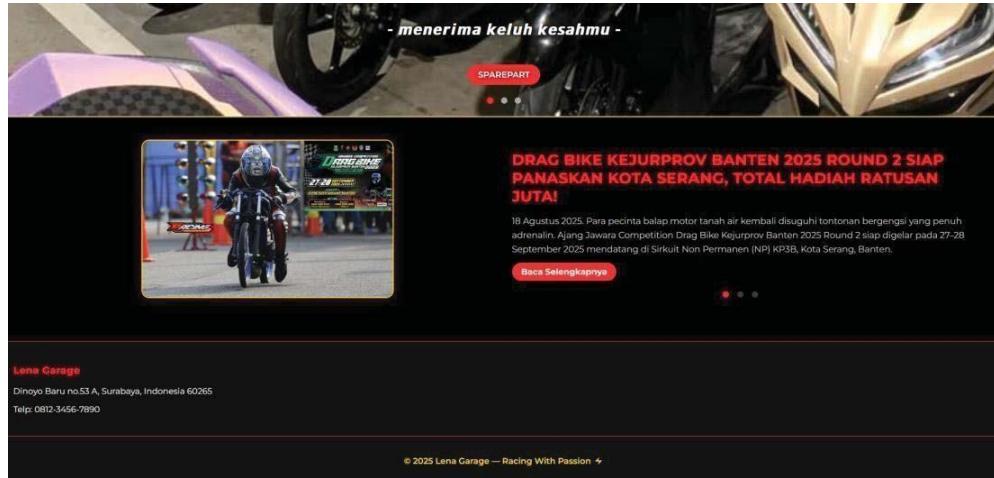
4.3.2 Halaman Utama

Halaman Utama merupakan halaman utama yang ditampilkan setelah pengguna memilih akses sebagai *user/admin*, berfungsi sebagai landing page yang menyajikan *informasi* umum tentang Lena Garage akses ke berbagai fitur yang

tersedia. Pada bagian atas halaman terdapat navigation bar yang menampilkan logo "LENA GARAGE" di sebelah kiri dan menu navigasi di sebelah kanan meliputi "Home", "Sparepart", "Hitung CC", tombol "Lokasi" berwarna merah, tombol "Admin" berwarna hijau, dan tombol "Logout" berwarna merah untuk keluar dari sistem. Bagian hero section menampilkan gambar banner sepeda motor dengan teks besar "LENA GARAGE" dan tagline "menerima keluh kesahmu" yang mencerminkan identitas dan layanan bengkel, serta terdapat tombol "SPAREPART" berwarna merah untuk mengarahkan pengguna langsung ke halaman katalog *sparepart*. Di bawah banner terdapat section berita atau artikel terbaru seperti informasi tentang *event drag bike* yang menunjukkan bahwa halaman ini juga berfungsi sebagai media informasi bagi pelanggan mengenai kegiatan atau berita terkait otomotif. Halaman ini dirancang dengan tampilan yang menarik menggunakan tema gelap (*dark theme*) dengan aksen warna merah yang konsisten dengan identitas brand Lena Garage, memberikan pengalaman pengguna yang informatif dan mudah dinavigasi untuk mengakses berbagai layanan yang ditawarkan oleh bengkel.



Gambar 4. 18 Halaman utama



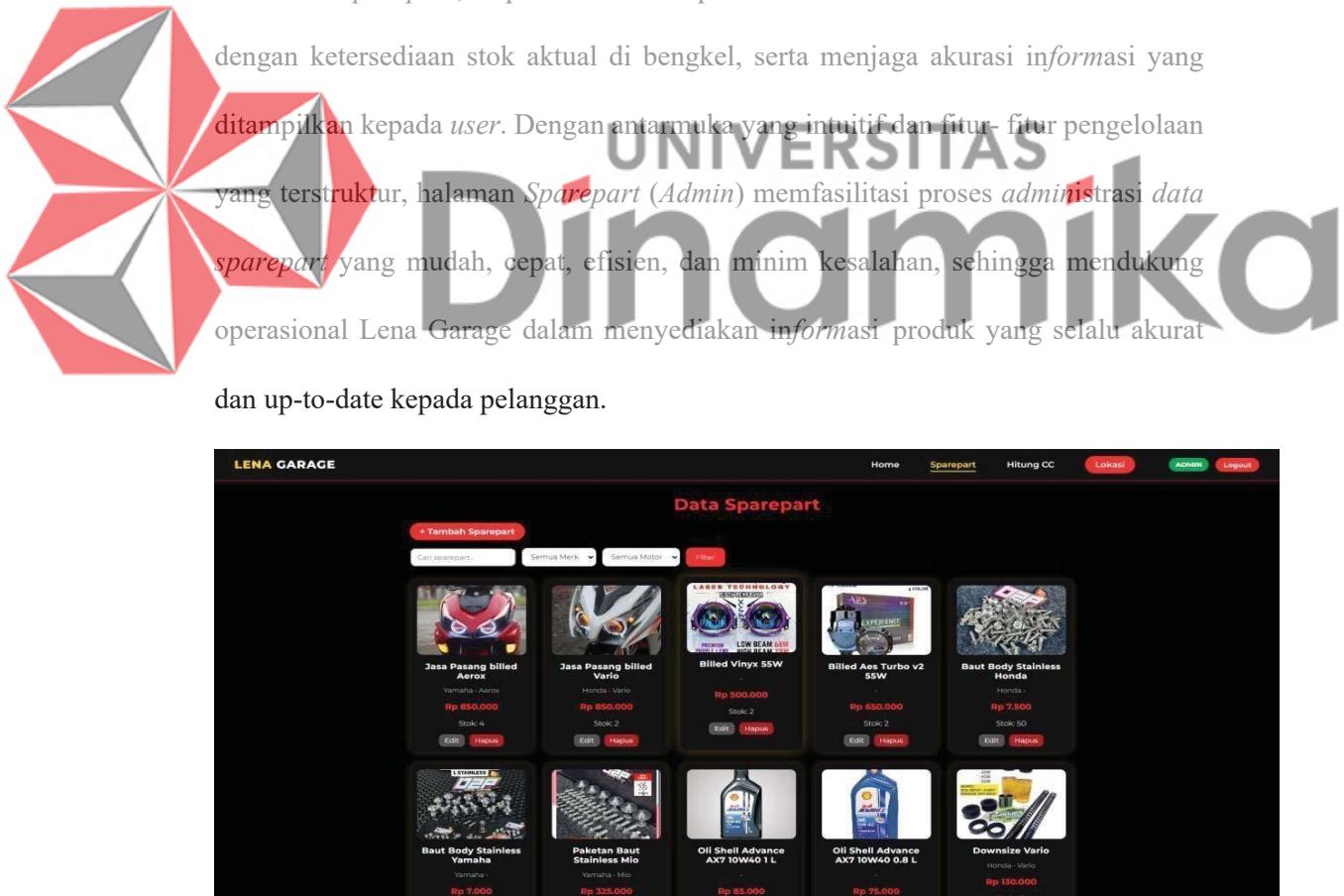
Gambar 4. 19 Halaman utama

4.3.3 Halaman *Sparepart (Admin)*

Pada bagian implementasi aplikasi, halaman *Sparepart (Admin)* merupakan antarmuka pengelolaan *data* yang dirancang khusus untuk *administrator* dalam mengelola seluruh informasi *sparepart* yang tersedia di Lena Garage secara komprehensif. Halaman ini menampilkan daftar *sparepart* dalam bentuk grid layout dengan kartu-kartu produk yang terorganisir rapi, dimana setiap kartu memuat informasi lengkap meliputi gambar *visual* produk, nama *sparepart*, merk dan jenis motor yang kompatibel, harga dalam *format rupiah*, serta jumlah stok yang tersedia. Berbeda dengan halaman *user* yang bersifat *read-only*, halaman *admin* ini dilengkapi dengan berbagai fitur manajemen *data* yang *powerful*, dimulai dari tombol "Tambah *Sparepart*" berwarna merah di bagian atas halaman yang memungkinkan *admin* untuk menambahkan produk *sparepart* baru ke dalam sistem dengan mengisi *form data* yang diperlukan. Untuk memudahkan pencarian dan pengelolaan *data* dalam jumlah besar, halaman ini juga menyediakan fitur

pencarian melalui *search bar* dan *filter* berdasarkan merk motor serta jenis motor yang dapat dikombinasikan untuk menemukan *sparepart* tertentu dengan cepat. Setiap kartu *sparepart* pada halaman *admin* dilengkapi dengan dua tombol aksi penting, yaitu tombol "Edit" berwarna biru yang memungkinkan *admin* untuk mengubah informasi *sparepart* seperti harga, stok, nama, atau spesifikasi lainnya, serta tombol "Hapus" berwarna merah yang berfungsi untuk menghapus *data sparepart* dari sistem ketika produk sudah tidak tersedia atau tidak lagi dijual.

Implementasi fitur CRUD (*Create, Read, Update, Delete*) yang lengkap pada halaman ini memastikan bahwa *admin* memiliki kontrol penuh terhadap manajemen inventaris *sparepart*, dapat melakukan pembaruan *data* secara *real-time* sesuai dengan ketersediaan stok aktual di bengkel, serta menjaga akurasi *informasi* yang ditampilkan kepada *user*. Dengan antarmuka yang intuitif dan fitur-fitur pengelolaan yang terstruktur, halaman *Sparepart (Admin)* memfasilitasi proses *administrasi data sparepart* yang mudah, cepat, efisien, dan minim kesalahan, sehingga mendukung operasional Lena Garage dalam menyediakan *informasi* produk yang selalu akurat dan up-to-date kepada pelanggan.



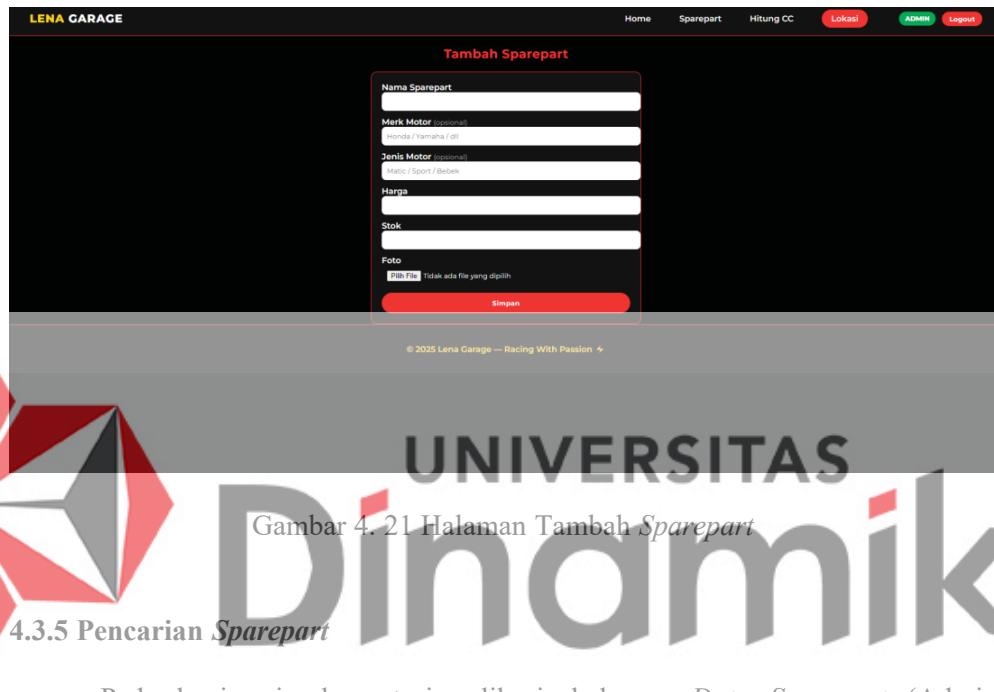
Gambar 4. 20 Halaman *sparepart* (*admin*)

4.3.4 Halaman Tambah Sparepart

Pada bagian implementasi aplikasi, halaman Tambah *Sparepart* merupakan antarmuka khusus administrator yang dirancang untuk menambahkan *data sparepart* baru ke dalam sistem inventaris Lena Garagesecara terstruktur dan efisien dengan *formulir input data* yang komprehensif. Halaman ini menampilkan judul "Tambah *Sparepart*" berwarna merah mencolok di bagian atas, diikuti dengan *formulir* yang dikemas dalam container abu-abu gelap dengan enam *field* input utama yang harus diisi, meliputi Nama *Sparepart* untuk mengidentifikasi produk, Merk Motor untuk menentukan kompatibilitas dengan brand kendaraan seperti Honda atau Yamaha, Jenis Motor untuk mengkategorikan tipe seperti Matic atau Sport, Harga dalam *format* numerik rupiah, Stok untuk mencatat jumlah unit yang tersedia, serta *field* Upload Foto yang dilengkapi dengan area *drag and drop* atau klik untuk memilih file gambar dengan spesifikasi maksimal 5MB dalam *format* PNG atau JPG.

Sistem validasi *real-time* terintegrasi pada setiap *field* untuk memastikan *data* yang diinput valid dan lengkap, dengan menampilkan pesan *error* berwarna merah di bawah *field* yang belum terisi atau terisi dengan *format* salah, seperti validasi *field* kosong, validasi numerik untuk harga yang harus berupa angka positif, validasi stok yang harus berupa bilangan bulat non-negatif, serta validasi upload foto yang memastikan file telah dipilih dengan *format* yang sesuai. Ketika admin mengunggah foto, sistem menampilkan *preview* gambar dengan opsi tombol hapus (ikon X) di pojok kanan atas untuk memungkinkan admin mengganti gambar jika diperlukan, dan setelah semua *data* tervalidasi dengan benar, admin dapat menekan tombol "Simpan" berwarna merah untuk menyimpan *data sparepart* baru ke

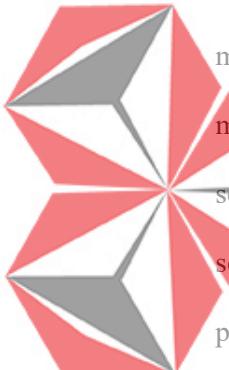
database atau tombol "Batal" berwarna abu-abu untuk membatalkan proses penambahan dengan konfirmasi peringatan bahwa *data* yang sudah diisi akan hilang, sehingga halaman ini memfasilitasi proses penambahan *data sparepart* yang mudah, terstruktur, dan minim kesalahan input dengan *user experience* yang intuitif dan responsif.



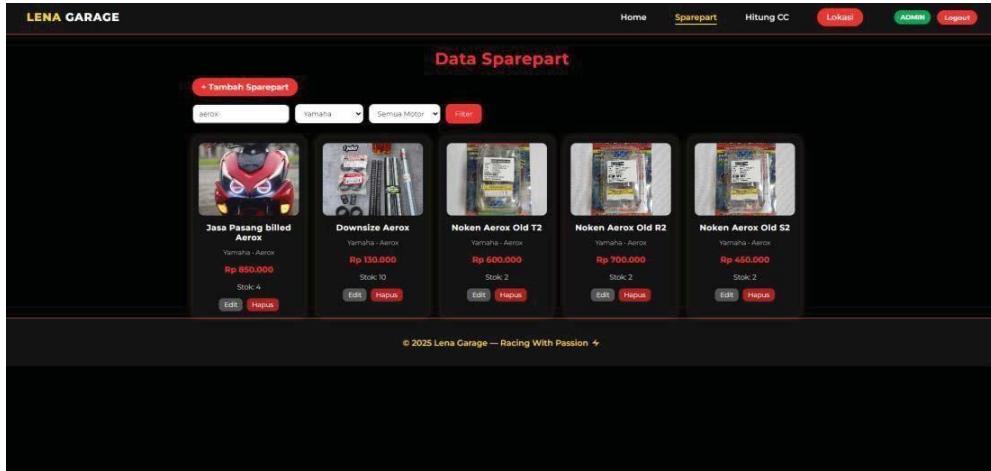
4.3.5 Pencarian *Sparepart*

Pada bagian implementasi aplikasi, halaman *Data Sparepart* (Admin) merupakan antarmuka pengelolaan inventaris yang menampilkan seluruh koleksi *sparepart* Lena Garage dalam format grid layout yang terorganisir dengan judul "*Data Sparepart*" berwarna merah mencolok di bagian atas halaman. Halaman ini dilengkapi dengan fitur pencarian dan *filtering* yang komprehensif di bawah tombol "+ Tambah Sparepart" berwarna merah, terdiri dari tiga komponen *filter* utama yaitu search bar dengan placeholder "Cari" yang memungkinkan admin mencari *sparepart* berdasarkan nama produk secara *real-time*, *dropdown filter* "Semua..." untuk menyaring berdasarkan merk motor seperti Honda, Yamaha, atau Suzuki, *dropdown filter* "Semua Motor" untuk menyaring berdasarkan jenis motor seperti

Matic, Sport, atau Bebek, serta tombol "Filter" berwarna merah yang berfungsi mengaplikasikan kombinasi *filter* yang dipilih untuk menampilkan hasil pencarian yang spesifik.



Sistem pencarian dan *filter* ini dirancang dengan algoritma yang responsif dan efisien, memungkinkan admin untuk menemukan *sparepart* tertentu dengan cepat di antara ratusan atau ribuan produk dalam *database* tanpa perlu scroll manual yang memakan waktu, dengan hasil pencarian yang ditampilkan secara instant dalam bentuk kartu produk yang memuat *informasi* lengkap meliputi foto produk, nama *sparepart*, merk dan jenis motor yang kompatibel, harga dalam *format rupiah*, jumlah stok yang tersedia, serta tombol aksi "Edit" berwarna biru dan "Hapus" berwarna merah untuk manajemen *data*. Fitur *filter* dapat dikombinasikan secara bersamaan, misalnya admin dapat *filter* *sparepart* untuk merk "Yamaha" dengan jenis "Aerox" sekaligus menggunakan search bar untuk mencari produk spesifik seperti "Noken", sehingga sistem akan menampilkan hanya produk yang memenuhi semua kriteria pencarian tersebut, dan ketika tidak ada *filter* yang diaplikasikan, halaman akan menampilkan seluruh *sparepart* yang tersedia dalam sistem secara default, dengan pagination di bagian bawah jika jumlah produk melebihi kapasitas satu halaman untuk menjaga *performa* loading dan *user experience* yang optimal



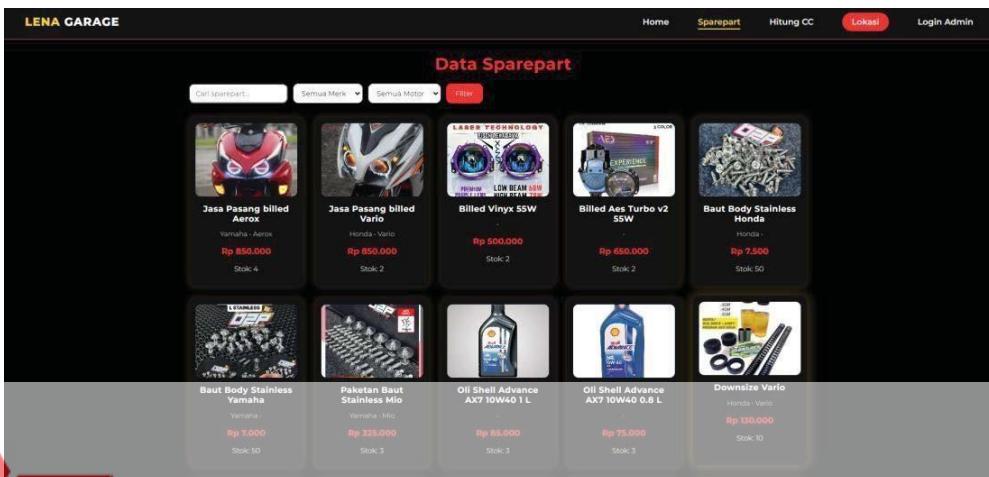
Gambar 4. 22 Pencarian *Sparepart*

4.3.6 Halaman *Sparepart (User)*

Pada bagian implementasi aplikasi, halaman *Sparepart (User)* berfungsi sebagai katalog digital yang menampilkan informasi lengkap mengenai daftar *sparepart* yang tersedia di Lena Garage untuk pengguna umum. Halaman ini dirancang dengan tampilan grid layout yang menampilkan kartu-kartu produk secara rapi dan responsif, dimana setiap kartu memuat informasi penting meliputi gambar produk, nama *sparepart*, merk dan jenis motor yang kompatibel, harga dalam format rupiah, serta ketersediaan stok.

Untuk meningkatkan kemudahan akses dan efisiensi pencarian, halaman ini dilengkapi dengan fitur pencarian melalui search bar dan fitur *filter* berdasarkan merk motor serta jenis motor yang dapat dikombinasikan untuk mempersempit hasil pencarian sesuai kebutuhan pengguna. Berbeda dengan halaman *admin*, pengguna pada halaman ini hanya memiliki akses baca (*read-only*) tanpa kewenangan untuk mengubah, menambah, atau menghapus *data sparepart*, sehingga integritas *data* tetap terjaga. Dengan desain visual yang menarik menggunakan background gelap dan highlight kontras, serta struktur informasi

yang terorganisir dengan baik, halaman ini berhasil menyajikan informasi *sparepart* yang mudah diaksessekaligus mendukung transparansi informasi stok dan harga secara *real-time* yang membantu pengguna dalam membuat keputusan pembelian yang lebih tepat sebelum berkunjung ke bengkel atau melakukan pemesanan.



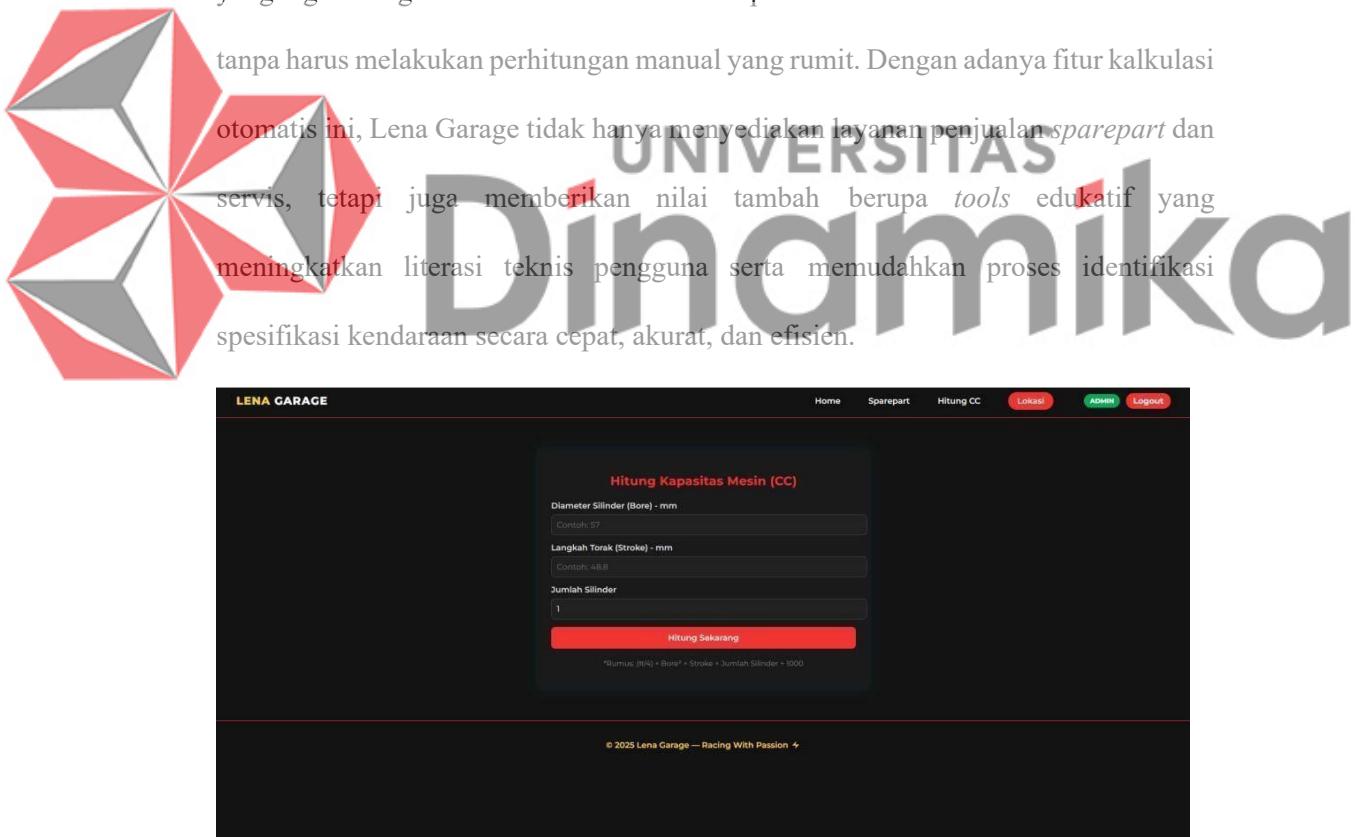
Gambar 4. 23 Halaman *sparepart* (User)

4.3.7 Halaman Hitung CC

Pada bagian implementasi aplikasi, halaman Hitung Kapasitas Mesin (CC) merupakan fitur utilitas yang dapat diakses oleh seluruh pengguna aplikasi, baik *admin* maupun *user*, yang berfungsi sebagai kalkulator otomatis untuk menghitung kapasitas mesin kendaraan bermotor. Halaman ini dirancang dengan antarmuka yang sederhana dan intuitif, menampilkan tiga kolom input yang harus diisi oleh pengguna, yaitu Diameter Silinder (*Bore*) dalam satuan milimeter yang merepresentasikan lebar lubang silinder mesin, Langkah Torak (*Stroke*) dalam satuan milimeter yang menunjukkan jarak pergerakan piston dari titik mati atas ke titik mati bawah, dan Jumlah Silinder yang menginformasikan berapa banyak silinder yang dimiliki mesin kendaraan tersebut. Setelah ketiga parameter tersebut

diinput dengan lengkap, pengguna dapat menekan tombol "Hitung Sekarang" yang berwarna merah untuk memproses perhitungan.

Sistem kemudian secara otomatis akan menghitung kapasitas mesin menggunakan rumus matematis standar perhitungan CC (*Cubic Centimeter*), yaitu $(\pi/4) \times \text{Bore}^2 \times \text{Stroke} \times \text{Jumlah Silinder} \div 1000$, dan menampilkan hasilnya dalam format yang mudah dibaca beserta penjelasan rumus yang digunakan di bagian bawah halaman. Fitur ini sangat bermanfaat bagi mekanik bengkel untuk memverifikasi spesifikasi mesin kendaraan yang *datang* untuk servis, bagi *admin* dalam mengelola *data sparepart* yang sesuai dengan kapasitas mesin tertentu, serta bagi pelanggan yang ingin mengetahui atau memverifikasi kapasitas mesin kendaraan mereka sendiri tanpa harus melakukan perhitungan manual yang rumit. Dengan adanya fitur kalkulasi otomatis ini, Lena Garage tidak hanya menyediakan layanan penjualan *sparepart* dan servis, tetapi juga memberikan nilai tambah berupa *tools* edukatif yang meningkatkan literasi teknis pengguna serta memudahkan proses identifikasi spesifikasi kendaraan secara cepat, akurat, dan efisien.



Gambar 4. 24 Halaman hitung cc

4.4 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode black box testing untuk memastikan setiap fitur berfungsi sesuai dengan kebutuhan. Pengujian dilakukan dengan menguji input dan output tanpa melihat struktur kode internal. Hasil pengujian dapat dilihat pada Tabel 4.4 berikut.

Tabel 4. 4 Pengujian Sistem

Fitur	Skenario	Hasil Test
Login	User memasukkan <i>username</i> dan <i>password</i> yang valid	Sukses
Login	User memasukkan <i>username</i> atau <i>password</i> yang salah	Sukses (menampilkan pesan <i>error</i>)
Login	User meninggalkan <i>field</i> kosong	Sukses (menampilkan pesan validasi)
Register	User mengisi semua <i>data</i> dengan format benar	Sukses
Register	User mengisi email dengan format salah	Sukses (menampilkan pesan <i>error</i>)
Register	User menggunakan <i>username</i> yang sudah terdaftar	Sukses (menampilkan pesan <i>error</i>)
Tambah Data	User mengisi <i>form</i> dengan <i>data</i> lengkap dan valid	Sukses
Tambah Data	User mengisi <i>form</i> dengan <i>data</i> tidak lengkap	Sukses (menampilkan pesan validasi)
Edit Data	User mengubah <i>data</i> yang sudah ada	Sukses
Edit Data	User mengosongkan <i>field</i> yang wajib diisi	Sukses (menampilkan pesan <i>error</i>)
Hapus Data	User menghapus <i>data</i> dengan konfirmasi	Sukses
Hapus Data	User membatalkan penghapusan <i>data</i>	Sukses (<i>data</i> tidak terhapus)
Search	User mencari <i>data</i> dengan <i>keyword</i> yang ada	Sukses (menampilkan hasil)
Search	User mencari <i>data</i> dengan <i>keyword</i> yang tidak ada	Sukses (menampilkan <i>data</i> kosong)
Logout	User menekan tombol <i>logout</i>	Sukses

dengan baik dan sesuai dengan skenario yang diharapkan.

Berdasarkan hasil pengujian pada Tabel 4.4, seluruh fitur telah berjalan



dengan baik dan sesuai dengan skenario yang diharapkan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan dan implementasi aplikasi Lena Garage, dapat disimpulkan bahwa aplikasi ini berhasil dibangun untuk membantu pengelolaan *data sparepart* secara terstruktur dan efisien. Aplikasi menyediakan fitur login *admin*, pengelolaan *data sparepart* (tambah, *edit*, hapus), pencarian dan *filter data*, serta fitur tambahan berupa perhitungan kapasitas mesin (CC) yang dapat diakses oleh *admin* dan *user*. Dengan adanya aplikasi ini, proses pengelolaan *data sparepart* menjadi lebih mudah, cepat, dan mengurangi risiko kesalahan pencatatan dibandingkan dengan sistem manual.

5.2. Saran

Untuk pengembangan selanjutnya, aplikasi Lena Garage dapat ditingkatkan dengan menambahkan fitur transaksi penjualan, laporan stok dan penjualan secara periodik, serta sistem hak akses yang lebih detail. Selain itu, tampilan antarmuka dapat terus disempurnakan agar lebih responsif di berbagai perangkat, dan integrasi dengan *database* yang lebih kompleks atau sistem pembayaran online dapat dipertimbangkan guna meningkatkan fungsionalitas dan kenyamanan pengguna.

DAFTAR PUSTAKA

Astutik, I. R. I., & Rosid, M. A. (2019). *Buku ajar pemrograman berbasis web* (1st ed., Vol. 1). UMSIDA Press.

Beaulieu, A. (2020). *Learning SQL: Generate, manipulate, and retrieve data* (3rd ed.). O'Reilly Media.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The unified modeling language user guide* (2nd ed.). Addison-Wesley Professional.

Duckett, J. (2014). *HTML and CSS: Design and build websites*. John Wiley & Sons.

Fowler, M. (2003). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley Professional.

Laudon, K. C., & Laudon, J. P. (2018). *Management information systems: Managing the digital firm* (15th ed.). Pearson.

Mozilla Developer Network. (2023). *HTML: HyperText Markup Language*. <https://developer.mozilla.org/en-US/docs/Web/HTML>

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Raharjo, B. (2015). *Modul pemrograman web (HTML, PHP, & MySQL)* (3rd ed.). Modula.

Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill Education.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.