

SKRIPSI

PEMBANGUNAN SISTEM PELAPORAN SPAREPART MENGUNAKAN BARCODE BERBASIS WEB STUDI KASUS BENGKEL MAJU MANDIRI SERVIS DAN AUTOPART



NAMA : UMAR RUSYDI BATUBARA
NIM : 1755201051

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI
RIAU
2021**

SKRIPSI

PEMBANGUNAN SISTEM PELAPORAN SPAREPART MENGUNAKAN BARCODE BERBASIS WEB STUDI KASUS BENGKEL MAJU MANDIRI SERVIS DAN AUTOPART



NAMA : UMAR RUSYDI BATUBARA

NIM : 1755201051

Diajukan sebagai persyaratan untuk mendapatkan Gelar Sarjana S1 Teknik Informatika

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI
RIAU
2021**

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi yang semakin maju membuat kinerja perusahaan relatif lebih cepat dan tepat untuk menghasilkan informasi yang dibutuhkan. Salah satunya, teknologi komputer yang dipakai dan diperlukan sebagai mempermudah kinerja dalam mencari informasi yang dibutuhkan perusahaan. Salah satu pemanfaatan teknologi informasi adalah perancangan sistem informasi persediaan secara komputerisasi, dimana sistem tersebut dapat memberikan informasi tentang data barang masuk dan keluar serta stok barang. Demi meningkatkan efisiensi dan keefektifan kerja, sistem informasi sangat dibutuhkan dalam suatu perusahaan. Perusahaan membutuhkan sistem yang memberikan kemudahan dalam menangani pemrosesan data transaksi penjualan. Bengkel Maju Mandiri Servis dan Autopart adalah jenis usaha yang melayani servis dan pembelian *sparepart* pada kendaraan mobil. Usaha ini memiliki kekurangan yaitu tidak mengetahui jumlah *sparepart* yang tersedia, karena tidak ada yang mencatat barang masuk dan barang keluar pada pelaporan stok *sparepart*, sehingga saat membutuhkan *sparepart*, stok *sparepart* ternyata sudah tidak ada. Hal ini membuat karyawan pergi mencari *sparepart* ke toko lain dan membuat konsumen menunggu dan memakan waktu yang cukup lama. Bengkel ini memiliki perangkat komputer yang hanya digunakan untuk melihat *cctv*. Oleh Karena itu, untuk lebih memanfaatkan kinerja komputer yang ada dibengkel tersebut, peneliti

merancang sistem informasi pelaporan menggunakan *Barcode* dengan metode berbasis web yang menggunakan sistem *database xampp*.

Barcode merupakan peran penting dalam melakukan pendataan *sparepart* dalam usaha, sehingga kinerja menjadi lebih cepat dan akurat. *Barcode* stiker ditempel disetiap *sparepart* yang dijual lalu stiker *Barcode* tersebut di *scan* menggunakan *Barcode Scanner* dan hasilnya akan langsung diproses oleh sistem.

Barcode Scanner merupakan alat yang berbentuk gagang dan berisi laser berwarna merah. Laser berfungsi untuk memindai *Barcode* yang biasanya terdapat pada kemasan suatu barang. *Barcode* tersebut berupa suatu kode berbentuk kotak-kotak atau garis-garis tebal vertikal yang kemudian diterjemahkan dalam bentuk angka-angka menggunakan *Barcode Scanner*.

Berdasarkan penjelasan maka dibuatlah **“Pembangunan Sistem Pelaporan Sparepart Menggunakan Barcode Berbasis Web Bengkel Maju Mandiri Servis dan Autopart”** dengan harapan dapat mempermudah melakukan pelaporan stok sparepart secara terkomputerisasi.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana sistem ini bekerja sehingga dapat mempermudah proses pendataan stok sparepart?
2. Bagaimana perbedaan pendataan secara *manual* dengan pendataan secara terkomputerisasi?

C. Batasan Masalah

Untuk membatasi ruang lingkup supaya fokus terhadap pembuatan sistem pelaporan stok barang maka dibatasi pada:

1. Sistem ini hanya untuk pendataan stok sparepart barang masuk dan keluar.
2. Sistem ini dibuat dalam bentuk web.

D. Manfaat Penelitian

1. Mempermudah melakukan pendataan stok barang masuk dan keluar.
2. Mempersingkat waktu pendataan stok *sparepart* secara terkomputerisasi dibandingkan pendataan stok barang secara manual.

E. Tujuan Penelitian

1. Mengetahui stok barang tersedia secara cepat dan akurat.
2. Mengetahui pelaporan secara terkomputerisasi lebih efisien dan modern dibandingkan pelaporan secara manual.

BAB II

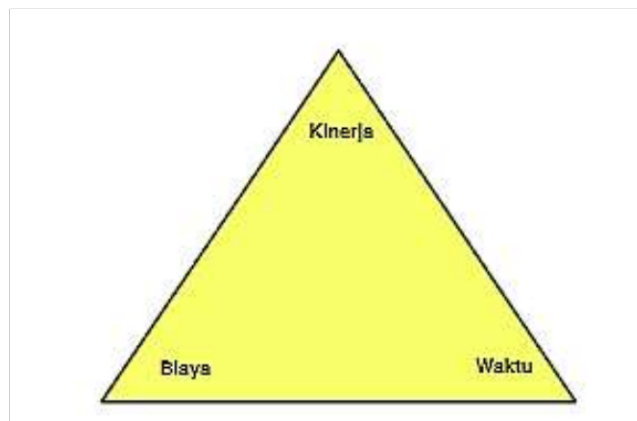
TINJAUAN PUSTAKA

A. Rekayasa Perangkat Lunak (*RPL*)

Rekayasa Perangkat Lunak (*RPL*) adalah kedisiplinan ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu menganalisa kebutuhan, menentukan spesifikasi dari kebutuhan, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.(R.Mulyanto, 2008).

Semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari *RPL*.(R.Mulyanto, 2008).

1. Tujuan Rekayasa Perangkat Lunak

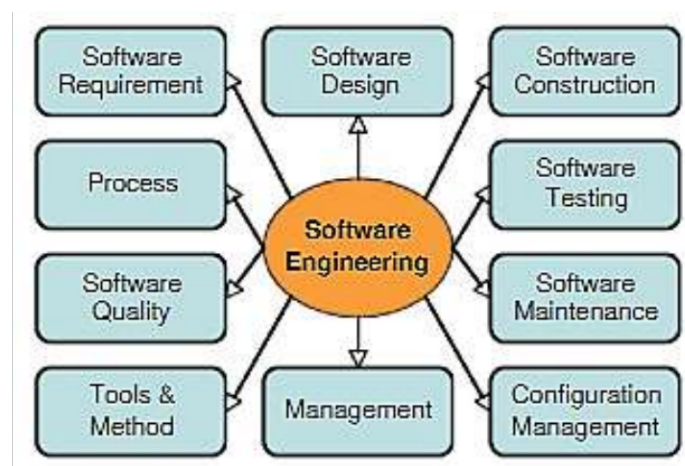


Gambar 2.1 Tujuan RPL
Sumber : Aunur R.Mulyanto, 2008

Bidang rekayasa selalu berusaha menghasilkan *output* yang kinerjanya tinggi, biaya rendah dan penyelesaian yang tepat waktu. Berikut adalah tujuan dari Rekayasa Perangkat Lunak (*RPL*) menurut R.Mulyanto (2008) adalah :

- a. Biaya produksi perangkat lunak rendah,
- b. Menghasilkan perangkat lunak yang memiliki kinerja tinggi, teruji dan tepat waktu,
- c. Menghasilkan perangkat lunak yang bekerja pada berbagai jenis platform,
- d. Menghasilkan perangkat lunak yang memiliki biaya perawatan rendah.

2. Ruang Lingkup Rekayasa Perangkat Lunak



Gambar 2.2 Ruang Lingkup RPL
Sumber : Aunur R.Mulyanto, 2008

Berikut adalah ruang lingkup Rekayasa Perangkat Lunak (*RPL*) menurut R.Mulyanto (2008) adalah :

- a. *Software requirements* berkaitan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak,
- b. *Software design* meliputi proses arsitektur, komponen, antarmuka, dan karakteristik lain dari perangkat lunak,

- c. *Software construction* berkaitan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian, dan pencarian kesalahan,
- d. *Software testing* meliputi pengujian keseluruhan perangkat lunak,
- e. *Software maintenance* berkaitan dengan perawatan saat perangkat lunak dioperasikan,
- f. *Software configuration management* berkaitan dengan perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu,
- g. *Software engineering management* berkaitan dengan pengelolaan dan pengukuran *RPL*, termasuk persiapan proyek perangkat lunak,
- h. *Software engineering tools and methods* mencakup analisis teoritis tentang alat bantu dan metode *RPL*,
- i. *Software engineering process* berhubungan dengan definisi, implementasi, pengukuran, pengelolaan, perubahan dan perbaikan *RPL*,
- j. *Software quality* memprioritaskan kualitas dan perkembangan perangkat lunak.

B. Bengkel

Bengkel adalah bangunan yang menyediakan ruang dan peralatan untuk melakukan konstruksi atau memperbaiki benda, perbengkelan adalah pengetahuan dan keterampilan tentang peralatan dan metode untuk membuat, membentuk, mengubah bentuk, merakit, ataupun memperbaiki benda menjadi

bentuk baru atau kondisi yang lebih baik secara manfaat maupun estetika. (Halim & Huwae, 2019).

C. *Database*

Database Management System (DBMS) berisi koleksi data yang saling berelasi dan set program untuk mengakses data. *DBMS* terdiri dari *Database* dan Set Program pengelola untuk menambah data, menghapus data, mengambil dan membaca data. (Ir.Harianto Kristanto, 1994).

Database adalah kumpulan *file-file* yang berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap *file* yang ada. Satu *database* memberitahukan satu kumpulan data yang dipakai dalam satu lingkup perusahaan, instansi. (Ir.Harianto Kristanto, 1994).

Dalam satu *file* terdapat *record-record* yang sejenis, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk memberitahukan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. (Ir.Harianto Kristanto, 1994).

Untuk menyebut isi dari *field* maka digunakan *attribute* atau merupakan judul dari satu kelompok *entity* tertentu, misalnya *attribute* alamat menunjukkan *entity* alamat mahasiswa. *Entity* adalah suatu obyek yang nyata dan akan direkam. (Ir.Harianto Kristanto, 1994).

Set program pengelola merupakan satu paket program yang dibuat untuk memudahkan pemasukkan informasi dan pengambilan informasi ke dalam *database*. (Ir.Harianto Kristanto, 1994).

1. *Entity*

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang Administrasi mahasiswa misalnya, *entity* adalah mahasiswa, buku, pembayaran, nilai test. Pada bidang kesehatan, *entity* adalah pasien, dokter, obat, kamar, diet.(Ir.Harianto Kristanto, 1994).

2. *Attribute*

Setiap *entity* memiliki *attribute* untuk mewakili suatu *entity*. Seorang mahasiswa dapat dilihat dari atributnya, misalnya nama, nomor mahasiswa, alamat, nama orang tua, hobi. *Attribute* juga disebut sebagai data elemen, data *field*, data *item*.(Ir.Harianto Kristanto, 1994).

3. *Data Value*

Data *value* adalah data aktual yang disimpan pada tiap data *attribute*. *Attribute* nama karyawan menunjukkan tempat informasi nama karyawan disimpan, sedangkan data *value* adalah Sutrisno, budiman, merupakan isi data nama karyawan tersebut.(Ir.Harianto Kristanto, 1994).

4. *Record*

Kumpulan elemen-elemen yang berkaitan menginformasikan suatu *entity* secara lengkap. Satu *record* mewakili satu data tentang seseorang misalnya, nomor karyawan, nama karyawan, alamat, kota, tanggal masuk.(Ir.Harianto Kristanto, 1994).

5. *File*

Kumpulan *record-record* yang memiliki panjang elemen dan *attribute* yang sama, namun data *valuenya* berbeda.(Ir.Harianto Kristanto, 1994).

Berikut kegunaan *Database* dalam penyusunan satu database yang digunakan untuk mengatasi masalah-masalah pada penyusunan pada data menurut Ir.Harianto Kristanto (1994), yaitu :

a. Redundansi dan inkonsistensi data

Jika *file-file* dan program diciptakan oleh programmer yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada *file-file* yang berbeda.

Contohnya alamat dan nomor telpon dari pelanggan kita tercatat pada *File Deposito* juga pada *file* Rekening Koran dan juga *file* Nasabah. Penyimpanan diberbagai tempat untuk data yang sama ini disebut sebagai redundansi dan mengakibatkan pemborosan ruang penyimpanan dan juga biaya untuk mengakses jadi lebih tinggi.

Penyimpanan data yang sama berulang-ulang dibeberapa file dapat mengakibatkan juga inkonsisten. Hal ini terjadi bila suatu ketika pelanggan tersebut pindah alamat dan nomor telponnya maka seharusnya ketiga *file* yang memuat data tersebut harus di *update*. Bila salah satu *file* mengandung data yang terlewat di *update*, maka terjadilah inkonsisten.

b. Kesulitan pengaksesan data

Pada suatu saat dibutuhkan untuk mencetak data siapa saja pelanggan yang berada di kode pos 55122 Yogyakarta, padahal belum tersedia program yang telah ditulis untuk mengeluarkan data tersebut. Maka kesulitan tersebut timbul, dan penyelesaian untuk itu adalah kearah *DBMS* yang mampu mengambil data secara langsung dengan bahasa yang *familiar* dan mudah digunakan (*user friendly*).

c. Isolasi data untuk standarisasi

Jika data tersebar dalam beberapa *file* dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka data dalam satu *database* dibuat satu format sehingga mudah membuat program aplikasinya. Dapat dibayangkan betapa sulitnya mebuat program aplikasi bila data dibuat dari format *textfilePascal*, *BASIC*, *C++* dan juga format dari versi *Lotus123* dan lainnya.

d. *Multipleuser* (banyak pemakai)

Dalam rangka mempercepat semua daya guna sistem dan mendapat responsi waktu yang cepat, beberapa sistem mengijinkan banyak pemakai untuk meng-*update* data secara *simultan*. Salah satu alasan mengapa *database* dibangun karena nantinya data tersebut digunakan oleh banyak orang dalam waktu yang berbeda, di akses oleh program yang sama tapi berbeda orang dan waktu.

e. Masalah keamanan (*security*)

Tidak setiap pemakai sistem *database* dibolehkan untuk mengakses semua data. Misalkan data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan dan *personalia*, tidak diperkenankan bagian gudang membaca dan mengubahnya.

Keamanan ini dapat diatur lewat program yang dibuat oleh pemogram atau fasilitas keamanan dari operating sistem misalnya *Novell Netware* untuk *Local Area Network*.

f. Masalah *integritas* (kesatuan)

Database berisi *file-file* yang saling berkaitan, masalah utama adalah bagaimana kaitan antara *file* tersebut terjadi. Meskipun kita mengetahui bahwa *file* A berkaitan dengan *file* B, namun secara teknis ada *field* kunci yang mengaitkan kedua *file* tersebut.

g. Masalah data *independence* (kebebasan data)

Pada suatu aplikasi yang kita buat menggunakan bahasa pemograman *BASIC* misalnya, bila program telah dibuat untuk menyelesaikan masalah pembacaan data untuk *file* pelanggan dengan *field* nomor, nama, alamat, maka setelah program tersebut haruslah diubah. Hal ini disebut bahwa program yang telah dibuat tidak bebas terhadap *database* yang ada.

Berlainan dengan paket bahasa yang diciptakan dari *DBMS*, apapun yang terjadi pada struktur *file*, setiap kali kita hendak melihat data cukuplah dengan *utility LIST*, hendak menambah data cukup

dengan *APPEND*. Ini berarti perintah-perintah dalam paket *DBMS* bebas terhadap *database*. Apapun perubahan dalam *database*, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

D. *MySQL*

MySQL merupakan salah satu *DBMS open source* yang paling populer pada saat ini. Meskipun dahulu *MySQL* pernah dikritisi karena tidak memiliki beberapa fitur yang ada dalam *DBMS* pada umumnya, namun saat ini *MySQL* sudah banyak dikembangkan.(Mustakim, 2018).

MySQL adalah sebuah server *database SQL multiuser* dan *multi-threaded*. *SQL* sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program server *database* ini adalah program *daemon* 'mysqld' dan beberapa program lain serta beberapa pustaka.(Mustakim, 2018).

MySQL adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak dipakai untuk berbagai kebutuhan. *MySQL* dikembangkan oleh *MySQLAB* Swedia. Hampir sebagian besar aplikasi *website* yang ada di internet dikembangkan menggunakan *MySQL* dan bahasa pemrograman lainnya, seperti *PHP*.(Enterprise, 2017).

Berikut ini hal-hal yang menyebabkan *MySQL* menjadi begitu populer. Menurut Enterprise (2017) :

1. Berlisensi *opensource*, sehingga dapat menggunakannya secara gratis,
2. Merupakan program yang *powerful* dan menyediakan fitur yang lengkap.
3. Menggunakan bentuk standar bahasa data *SQL*,

4. Dapat bekerja dengan banyak sistem operasi dan dengan bahasa-bahasa pemrograman, seperti *PHP*, *PERL*, *C*, *C++*, *JAVA*, dan lain-lain,
5. Bekerja dengan cepat dan baik, bahkan dengan data set yang banyak,
6. Sangat mudah digunakan dengan *PHP* untuk pengembangan aplikasi web,
7. Mendukung banyak *database*, sampai 50 juta baris atau lebih dalam suatu table,
8. Dapat dikustomisasi sesuai dengan keinginan,
9. Mendukung penuh terhadap kalimat *SQL GROUP BY* dan *ORDER BY*.
Mendukung terhadap fungsi penuh (*COUNT()*, *COUNT()*, *DISTINCT()* *AVG()*, *STD()*, *SUM()*, *MAX()* dan *MIN()*).

E. *Xampp*

XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi web server *Apache*, *PHP* dan *MySQL* secara *manual*. *XAMPP* akan menginstallasi dan mengkonfigurasi secara *otomatis* atau *auto* konfigurasi. (Content & Informatika, 2015).

F. *Web*

WWW atau *World Wide Web* atau web merupakan sistem yang saling terkait dalam sebuah dokumen berformat *hypertext* yang berisi beragam informasi, baik tulisan, gambar, suara, video, dan informasi multimedia lainnya dan dapat diakses melalui sebuah perangkat yang disebut web *browser*. Untuk menterjemahkan dokumen dalam bentuk *hypertext* ke dalam bentuk dokumen yang bisa dipahami, maka web *browser* melalui web *client*

akan membaca halaman web yang tersimpan di sebuah web server melalui protokol yang biasa disebut *http* atau *Hypertext Transfer Protocol*. (Morita et al., 1974).

1. Perkembangan *Web*

a. *Web 1.0*

Menurut Prasetiadi (2011), istilah *Web 1.0* merupakan retronim, maksudnya istilah ini tidak ada sebelum era *Web 2.0 booming* pada tahun 2004. Ciri-ciri *Web 1.0* adalah sebagai berikut :

1. Web pages bersifat statik, di mana *website* hanya bersifat “*read-only*” sehingga terjadi komunikasi satu arah. Pembaca hanya dapat menerima informasi, tanpa berpartisipasi dalam memberikan informasi,
2. Penggunaan *frameset*, yaitu tampilan beberapa buah *webpage* dalam satu *window*,
3. Penggunaan *tag HTML* seperti *<blink>* dan *<marquee>*,
4. Adanya fitur buku tamu online,
5. Penggunaan *button* dengan format *GIF*, biasanya 88 x 31 untuk promosi produk.

b. *Web 2.0*

Menurut Prasetiadi (2011), istilah *Web 2.0* diperkenalkan oleh *Darcy DiNucci* dalam artikelnya yang berjudul “*Fragmented Future*”, yaitu : “*The Web we know now, which loads into a browser window in essentially static screenfulls, is only an embryo of the Web*

to come. The first glimmerings of Web 2.0 are beginning to appear, and we are just starting to see how that embryo might develop. The Web will be understood not as screenfulls of text and graphics but as a transport mechanism, the ether through which interactivity happens. It will appear on your computer screen, on your TV se your car dashboard your cell phonehand-held game machines maybe even your microwave oven.”

Meskipun demikian, istilah *Web 2.0* mulai muncul ke permukaan ketika Tim *O'Reilly* dari *O'Reilly Media* mengadakan konferensi *Web 2.0* pertama kalinya di tahun 2004. Pada hakikatnya, *Web 2.0* menggambarkan aplikasi *web* yang memungkinkan kita untuk berbagi informasi, *interoperability*, *user-centered design*, dan berkolaborasi. Intinya, *user* dimungkinkan untuk berkontribusi terhadap konten informasi yang tersedia dalam *web*. Hal ini berbeda dengan *Web 1.0*, dimana *user* hanya menjadi pengguna pasif yang hanya menelan informasi pada *website* tertentu. Berikut ini adalah beberapa karakteristik *Web 2.0* Menurut Prasetiadi (2011) :

1. Pengguna *web* dapat mengubah isi *webpage*, seperti pada *Wikipedia*,
2. Penggunaan *webpage* untuk berhubungan dengan *user* lain, seperti pada situs *Facebook* atau *MySpace* yang *booming* karena pengguna mudah untuk menjalin silaturahmi dengan rekannya,

bahkan dengan teman-teman lama yang sudah lama tidak dijumpai,

3. *Share Content* yang cepat dan efisien, seperti *YouTube*, dimana member dapat membuat video dan mengunggahnya, lalu orang lain dapat melihat video tersebut dalam waktu yang relatif cepat setelah pengunggahan video,
4. Cara baru untuk mendapatkan informasi, yaitu *Really Simple Syndication (RSS)*. *user* dapat menerima *notifikasi* apabila suatu *web page* ter-update,
5. Koneksi internet tidak terbatas melalui komputer, tetapi juga dapat melalui perangkat lain, seperti telepon seluler.

c. *Web 3.0*

Dengan boomingnya *Web 2.0*, para pakar internet mulai berpikir untuk mengembangkan generasi selanjutnya dari *Web 2.0*, yaitu *Web 3.0*. Konsep *Web 3.0* pertama kali diperkenalkan pada tahun 2001, yaitu saat *Tim Berners-Lee*, penemu *World Wide Web*, menulis sebuah artikel ilmiah yang menggambarkan *Web 3.0* sebagai sebuah sarana bagi mesin untuk membaca halaman *Web*. Hal ini berarti bahwa mesin akan memiliki kemampuan membaca *Web* sama seperti manusia.

Beberapa pakar berpendapat bahwa *browser* dapat menjadi asisten pribadi *user* yang mengetahui segala sesuatu tentang *user* tersebut dan dapat mencari berbagai informasi di internet untuk menjawab

pertanyaan yang kita ajukan. Dalam *Web 3.0*, internet digunakan untuk membuat koneksi dengan informasi, sementara di *Web 2.0* biasanya internet digunakan untuk berkoneksi dengan teman kita. Berikut ini adalah beberapa gambaran tentang *Web 3.0*.

1. Dengan menggunakan teknologi *web (search engine)*, kita dapat mencari lokasi, penginapan, serta transportasi, Pencarian ini dilakukan oleh internet. Browser menganalisis data-data dan menampilkan hasil analisis ke *user*. Hal ini tentu akan lebih praktis,
2. *Search engine* dapat menganalisis konteks dari *keyword* yang dimasukkan,
3. Setiap *user* mempunyai profil internet yang unik berdasarkan catatan kebiasaan ketika mengakses internet. *Web 3.0* memanfaatkan *history* ini ketika melakukan *searching*. Artinya, ketika dua orang yang berbeda melakukan *searching* dengan *keyword* yang sama, hasil yang ditampilkan kemungkinan dapat berbeda, sesuai dengan rekaman *browsing* masing-masing pengguna internet.

2. Web Server

Web Server atau *Server Web* merupakan sebuah perangkat lunak server yang berfungsi menerima permintaan dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* melalui *protocol HTTP* atau *HTTPS*. *Web server*

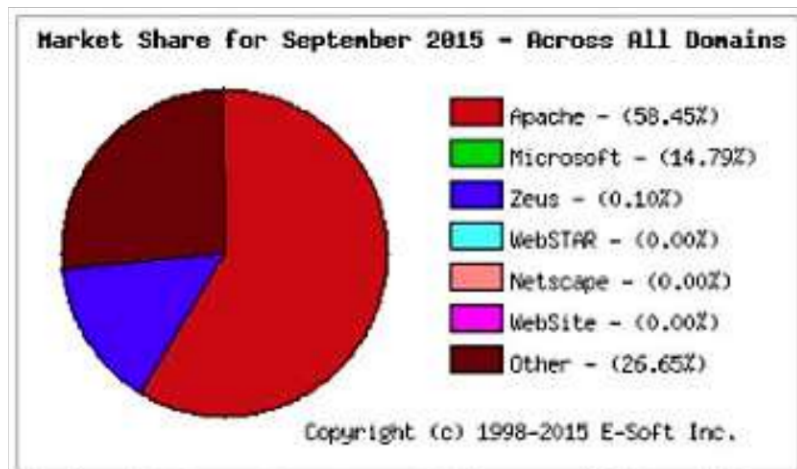
bertugas mengelola halaman-halaman *web* dan dokumen-dokumen lainnya. Pada dasarnya *web* server hanyalah sebuah perangkat lunak yang bisa saja berada di komputer apapun dengan spesifikasi apapun. PC biasapun dapat berfungsi sebagai *web* server. (Solichin, 2016)

Berikut ini beberapa keunggulan menurut Solichin (2016), dari perangkat server *web apache* :

- a. Gratis. *Apache* dapat diunduh dan dipergunakan secara gratis.
- b. Berjalan diberbagai sistem operasi. *Apache* dapat dipasang dan dijalankan di hampir semua sistem operasi seperti *windows*, *Linux*, *FreeBSD*, *Mac OS* dan berbagai sistem operasi turunan *Unix*.
- c. Populer. Berdasarkan survey dari *Netcraft*, pada bulan September 2015 sebanyak 50% situs web aktif di internet menggunakan *web* server *Apache*. 58,45% domain di internet menggunakan *web* server *Apache*. Besarnya pengguna *Apache* berimplikasi langsung dengan besarnya dukungan komunitas. Hal tersebut cukup penting terutama jika kita menghadapi masalah dan kesulitan dengan perangkat server *web* yang digunakan.



Gambar 2.3 Survey Penggunaan Webservice oleh Netcraft
Sumber : Achmad Solichin, 2016



Gambar 2.4 Survey Penggunaan Webservice oleh SecuritySpace
Sumber :Achmad Solichin, 2016

- d. Tangguh. Setelah *Apache* berhasil dipasang dan berjalan di suatu komputer, maka jarang sekali terjadi masalah seperti rusak atau *corrupt*. *Apache* akan tetap berjalan selama komputer masih hidup.
- e. Mudah dikustomisasi. *Apache* sangat mudah dikonfigurasi sesuai kebutuhan. Penambahan modul juga memungkinkan untuk dilakukan dengan mudah.
- f. Aman. *Apache* mendukung penerapan *protocolSSL* sehingga *web* akan lebih aman.

3. Peramban *Web*(*Web Browser*)

Peramban *web* atau dikenal dengan *web browser* merupakan perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di internet. *Browser* digunakan untuk menampilkan halaman-halaman *web*. Seluruh komponen *web* termasuk teks, gambar dan komponen lain yang dibangun dengan teknologi *client-side scripting* dapat menampilkan di *web browser*.(Solichin, 2016).

4. *HTML, CSS dan Javascript*

HTML merupakan singkatan dari *Hypertext Markup Language*. *HTML* dikembangkan pertama kali oleh tim *Berners-Lee* bersamaan dengan *protocol HTML (Hypertext Transfer Protocol)* pada tahun 1989. Tujuan utama pengembangan *HTML* adalah untuk menghubungkan satu halaman *web* dengan halaman *web* lainnya. Tentunya pada awal perkembangannya, halaman *web* hanya berupa teks, tidak seperti sekarang.(Solichin, 2016).

Pada dasarnya, setiap halaman *web* ditulis dalam bentuk *HTML*. *HTML* merupakan bahasa pemrograman web yang memberitahukan peramban *web (web browser)* bagaimana menyusun dan menyajikan konten dihalaman *web*. Dengan kata lain, *HTML* adalah pondasi *web*. *HTML* disusun dengan bahasa yang sederhana, sehingga sangat mudah diimplementasikan. Saat ini, *HTML* dapat menampilkan obyek-obyek seperti teks, table, tautan, gambar, audio, dan video.(Solichin, 2016).

HTML merupakan bahasa dasar *web* yang berfungsi untuk menampilkan berbagai komponen *web*. Sementara itu, untuk mempercantik tampilan *web*, dikembangkanlah *CSS* atau *Cascading Style Sheet*. *CSS* pertama kali diusulkan oleh Hakon Wium Lie pada tahun 1994 dan selanjutnya distandarisasi oleh *W3C*. *CSS* memberikan cara yang mudah dan efisien bagi pemrogram untuk menentukan tata letak halaman *web* dan mempercantik halaman dengan elemen desain seperti warna, sudut bulat, gradient, dan animasi.(Solichin, 2016).

Selain *HTML* dan *CSS*, sebuah aplikasi berbasis *web* tidak dapat dilepaskan dari teknologi *Javascript*. Pertama kali *Javascript* dikembangkan oleh *Netscape* dengan nama awal *LiveScript*. Fungsi utama dari *Javascript* adalah untuk menambah fungsionalitas dan kenyamanan halaman *web*. *Javascript* lebih fokus pada proses pengolahan data di sisi *client* serta menyajikan komponen *web* yang lebih interaktif. *Javascript* makin populer sejak kemunculan *AJAX* (*Asynchronous Javascript and XML*) yang memungkinkan interaksi antara *client* dan server lebih elegan dan fleksibel.(Solichin, 2016).

5. *PHP*

PHP merupakan bahasa (*scripting language*) yang ditempelkan (*embedded*) ketika digunakan dalam halaman *web*. Maksudnya adalah bahwa kode *PHP* ditempelkan di dalam kode *html* (Rulianto kurniawan, 2010). *Software PHP* bekerja bersama dengan *Web Server*. *Web Server* adalah *software* yang mengirim halaman *web* kepada dunia.(Sidratul munti, 2019).

PHP merupakan perangkat lunak *open source*, yang mana penulisan kode program *PHP* menyatu dengan *HTML* yang berjalan pada *server-side*. Hal ini berarti semua sintak yang telah ditulis akan sepenuhnya dijalankan pada server; hanya hasil dari *script* yang telah dieksekusi saja yang dikirimkan pada sisi *client/browser* tanpa bisa mengetahui kode yang digunakan. *PHP* merupakan bahasa pemrograman yang sesuai untuk membuat aplikasi *website* dinamis seperti *CMS* karena memiliki

performa yang tinggi, mudah dipelajari, *multiplatform*, aman, *open source*, serta mudah dikoneksikan dengan berbagai macam *database*.(Sidratul munti, 2019).

PHP adalah sebuah bahasa *scripting* yang terpasang pada *HTML*.Sebagian besar sintaks mirip dengan bahasa *C*, *Java* dan *Perl*, ditambah beberapa fungsi *PHP* yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancangan *web* menulis halaman *web* dinamik dengan cepat.(Mustakim, 2018).

PHP merupakan salah satu bahasa pemrograman berbasis *web* untuk pengembang *web*. *PHP* pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *software* dan anggota tim *Apache*, dan dirilis pada akhir tahun 1994. *PHP* dikembangkan dengan tujuan awal hanya untuk mencatat pengunjung pada *website* pribadi Rasmus Lerdorf. Pada rilis keduanya, ditambahkan *Form Interpreter*, sebuah *tools* untuk melakukan penerjemahan perintah *SQL*. Rilis kedua disebut dengan *PHP/FI*. Sejak itu, *PHP* mulai diterima sebagai sebuah bahasa pemrograman baru yang sangat diminati. Terbukti pada pertengahan tahun 1997, tercatat sekitar 50.000 situs diseluruh dunia telah menggunakan *PHP*.(Solichin, 2016).

Dengan bertambah banyaknya pengguna *PHP* di seluruh dunia, maka *PHP* tidak memungkinkan lagi untuk dikelola oleh satu orang saja. Sehingga dibentuk sebuah tim pengembang yang mengorganisir kontibutor dari seluruh dunia dengan model pengembangan proyek *open*

source “*benevolent junta*”. Tim tersebut dipimpin oleh dua orang programmer, Zeev Suraski dan Andi Gutmans. Keduanya lalu mendirikan sebuah perusahaan *PHP* dengan nama *Zend* (akronim dari nama Zeev Suraski dan Andi Gutmans). Selanjutnya *Zend* merilis versi *PHP3* dan *PHP4*.(Solichin, 2016).

Tahun 1998 terjadi peningkatan pengguna *PHP* yang sangat besar, bersamaan dengan naiknya popularitas penggunaan teknologi *open source*. Pada bulan oktober 1998, *PHP* sudah digunakan oleh sekitar 100.000 domain di internet. Setahun setelahnya, sudah mencapai 1 juta domain. Perkembangan selanjutnya sangatlah luar biasa. Pada tahun 2004 tercatat sekitar 15 juta server *web* terpasang *PHP* di dalamnya. Berdasarkan survey yang dilakukan oleh *Netcraft*, pada bulan januari 2013 situs yang menggunakan *PHP* sudah mencapai 244 juta situs.(Solichin, 2016).

Seiring dengan waktu, *PHP* terus dikembangkan baik dari sisi fitur, keamanan, maupun performanya. Versi terakhir *PHP* adalah *PHP 5.6.14* (Oktober 2015). Sejak *PHP5*, terjadi penambahan dan peningkatan fitur yang cukup signifikan. *PHP5* dibangun dengan konsep berorientasi obyek (*OO*) secara penuh seperti halnya bahasa pemrograman seperti *Java* dan *C++*.Versi sebelumnya memang *PHP* belum sepenuhnya mendukung konsep *OO*. Mulai *PHP5*, seluruh fungsi bawaan *PHP* ditulis ulang menggunakan konsep *OO*, walaupun pada masa transisi *PHP*

masih mendukung gaya klasik yaitu *procedural*. Sedikit demi sedikit gaya penulisan *procedural* akan dihilangkan dari *PHP*.(Solichin, 2016).

Beberapa keunggulan *PHP* terangkum dalam daftar berikut ini menurut Solichin (2016) :

- a. Gratis. *PHP* dapat diunduh dan dipergunakan secara gratis. *PHP* dikembangkan oleh komunitas *open source* dan akan selalu didistribusikan secara gratis,
- b. *PHP* berlisensi *GNU General Public License (GPL)*. Hal tersebut menjadi jaminan bahwa semua versi *PHP* akan selalu didistribusikan secara gratis. Walaupun demikian, untuk setiap program dan aplikasi yang dihasilkan menggunakan *PHP*, pengembang dapat menentukan lisensinya sendiri (tidak harus berlisensi terbuka). Selain itu, dalam proses distribusinya, tetap diperkenankan menarik biaya dalam jumlah yang wajar, misalnya sebagai biaya *packaging* atau *burning* ke *CD ROM*,
- c. Performa Handal. *PHP* sangatlah efisien, dengan sebuah spesifikasi server yang tidak mahal, dapat melayani jutaan akses per hari,
- d. Dukungan *Database*. *PHP* mendukung hampir semua perangkat *database*, mulai dari *MySQL*, *Oracle*, *PostgreSQL*, *Informix*, *Interbase*, *Sybase*, *MariaDB* hingga *SQLite*. Bahkan beberapa diantaranya sudah terhubung dengan *PHP* secara *native (default)*,
- e. Pustaka Bawaan. *PHP* dibangun secara khusus untuk aplikasi berbasis *web*. Oleh karena itu, disediakan banyak sekali pustaka

- (*library*) bawaan berkaitan dengan *web* yang dapat langsung digunakan. Sebagai contoh, *PHP* memiliki pustaka untuk menampilkan *PDF on-the-fly* (langsung di *browser*), membuat grafik *on-the-fly*, mengirim *email*, mengelola *file*, kompresi, enkripsi, penanganan *session* dan sebagainya,
- f. *Cross platform*. *PHP* dapat dijalankan dengan baik pada hampir semua Sistem Operasi. *PHP* dapat berjalan di *Linux*, *Unix*, *Windows*, *Mac OS*, *FreeBSD*, *Sun Solaris* dan bahkan saat ini dapat berjalan di sistem operasi *Android* melalui proyek *DroidPHP*,
 - g. Mudah Dipelajari. Perintah-perintah *PHP* sangat mudah dipelajari karena sebagian besar perintah *PHP* diadopsi dari bahasa pemrograman populer seperti *Bahasa C/C++*, *Java* dan *perl*.

G. Hubungan PHP dengan HTML

Halaman *web* biasanya disusun dari kode-kode *html* yang disimpan dalam sebuah *file* berekstensi *.html*. *Filehtml* ini dikirimkan oleh server ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan pemograman *web*, pemograman ini harus diterjemahkan oleh *web* server sehingga menghasilkan kode *html* yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode *html* sehingga dapat langsung ditampilkan bersama dengan kode-kode *html* tersebut. Program *php* ditambahkan dengan mengapit program tersebut di antara tanda `<?php` dan `?>`. (Mustakim, 2018).

Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode *html*. *File html* yang telah dibubuhi program *PHP* harus diganti ekstensinya menjadi *.php3* atau *.php*. *PHP* merupakan bahasa pemrograman web yang bersifat *server-side HTML (Embedded Scripting)*, dimana *script*-nya menyatu dengan *HTML* dan berada di server. Artinya sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan diserver tetapi disertakan *HTML* biasa. *PHP* dikenal sebagai bahasa *scripting* yang menyatu dengan *tag HTML*, dieksekusi di server dan digunakan untuk membuat halaman *web* yang dinamis seperti *ASP (Active Server Pages)* dan *JSP (Java Server Pages)*.(Mustakim, 2018).

H. Barcode

Barcode adalah Kode yang dapat dibaca mesin dalam bentuk angka dan pola garis paralel dengan lebar yang berbeda-beda, dicetak untuk *identifikasi* suatu produk secara unik. Secara umum penggunaan sistem *Barcode* telah memberikan banyak manfaat bagi bisnis dan organisasi karena mampu memberikan peningkatan dalam produktivitas dan efisiensi yang signifikan bila dibandingkan dengan sistem *manual*. Dalam kegiatan bisnis dan organisasi sistem *Barcode* digunakan dalam melacak produk, harga, status dan tingkat stok untuk manajemen terpusat dalam sebuah sistem perangkat lunak komputer yang bisa diakses informasinya secara *real time* dan bisa terhubung secara *global*. (Mulyono, 2019).

Barcode atau kode batang adalah suatu kumpulan data optik yang dibaca mesin. Sebenarnya, kode batang ini mengumpulkan data dalam lebar (garis)

dan jarak garis paralel dan dapat disebut sebagai kode batang atau *simbologilinear* atau *1D* (1 dimensi). Tetapi juga memiliki bentuk persegi, titik, *heksagon* dan bentuk geometri lainnya di dalam gambar yang disebut kode matriks atau *simbologi 2D* (2 dimensi). Selain tak ada garis, sistem *2D* sering juga disebut sebagai kode batang. (Sarmidi, 2018).

Kode batang atau *Barcode* dibaca mesin yang terdiri atas susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, sangat sederhana tetapi sangat berguna untuk menyimpan data-data spesifik misalnya kode produksi, tanggal kadaluwarsa, dan nomor identitas dari suatu produk. (Mulyono, 2019).

Penggunaan *Barcode 1D* hanya dapat menampung maksimal 85 karakter, maka seiring dengan perkembangan teknologi pemindai maka saat ini terjadi penurunan kebutuhan dan permintaan untuk *Barcode 1D* yang berbanding terbalik dengan tingkat permintaan *2D Barcode*. (Mulyono, 2019).

Pengembangan teknologi *2D Barcode* meningkatkan performansi dalam hal menyimpan lebih dari 7.000 karakter, memungkinkan untuk mengirimkan hampir dua paragraf informasi, dengan penggunaan ke *Barcode 2D*, kegiatan bisnis yang kompleks dapat lebih terwakili karena dapat menyampaikan informasi yang jauh lebih rumit dan banyak, seperti tanggal mulai dan selesainya kegiatan operasi, kadaluwarsa, nomor seri, dan hal lainnya tanpa perlu pemindaian tambahan. (Mulyono, 2019).

1. Barcode 1 Dimensi (*Linier barcode*)

a. Code 39 (*code 3 of 9*)



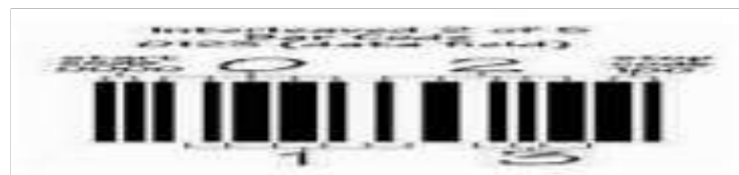
Gambar 2.5 Contoh Barcode 39
Sumber : Sarmidi, Sidik Ibnu Rahmat 2018

b. Code 128



Gambar 2.6 Contoh Barcode 128
Sumber : Sarmidi, Sidik Ibnu Rahmat 2018

c. Interleaved 2 of 5



Gambar 2.7 Contoh Barcode Interleaved 2 of 5
Sumber : Sarmidi, Sidik Ibnu Rahmat 2018

d. UPC (*Universal Prouct Code*)



Gambar 2.8 Contoh Barcode UPC
Sumber : Sarmidi, Sidik Ibnu Rahmat 2018

2. Barcode 2 Dimensi



Gambar 2.9 Contoh Barcode 2 Dimensi
Sumber : Sarmidi, Sidik Ibnu Rahmat 2018

I. UML

Unified Modelling Language (UML) adalah penerus dari *object-oriented analysis and design (OOA&D) methods* yang muncul pada akhir 80an dan awal 90an. *UML* secara langsung menggabungkan *methods* dari *Booch*, *Rumbaugh (OMT)*, dan *Jacobson*. *UML* berkembang melalui sebuah proses standarisasi yang dilakukan oleh *OMG (Object Management Group)* (Anonymous, 2008). *UML* disebut sebagai bahasa untuk permodelan, bukan sebuah *method*. Hampir semua *methods* mengandung, paling tidak dalam beberapa prinsip, dua dari sebuah bahasa permodelan dan sebuah proses. Bahasa permodelan tersebut (terutama yang berbasis grafis) adalah sebuah notasi yang menggunakan *methods* untuk mengekspresikan sebuah rancangan. Proses tersebut adalah tuntunan yang dilakukan dalam setiap langkah untuk merancang sesuatu. (Novianti & Fauziah, 2009).

Secara umum *UML (Unified Modelling Language)* merupakan bahasa untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. *UML* ini digunakan oleh para pengembang sebagai sarana untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna suatu sistem atau perangkat lunak. *UML* menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive system*) (Nugroho, 2005). Cara kerja *UML* adalah dengan mendefinisikan notasi dan sebuah meta-model sehingga pembacaan alur dalam melakukan *forward* maupun *reverse engineering*. (Novianti & Fauziah, 2009).

1. *Use Case Diagram*

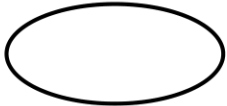




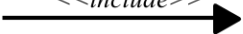
Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan *system* untuk melakukan pekerjaan-pekerjaan tertentu. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. (Agung et al., 2018).

Menurut Agung et al. (2018), *Use Case Diagram* dapat digunakan untuk :

- a. Menyusun *requirement* sebuah sistem,
- b. Mengkomunikasikan rancangan dengan klien,
- c. Merancang *test case* untuk semua *feature* yang ada pada sistem.

Berikut adalah simbol-simbol yang ada pada *Use Case Diagram*
Menurut Heriyanto (2018) :

Tabel 2.1 Simbol pada *Use Case Diagram*

SIMBOL	DESKRIPSI
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
<p>Aktor</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.
<p>Asosiasi</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
<p>Ekstensi <<extends>></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.
<p>Generalisasi</p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Include <<include>></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber : Yunahar Heriyanto, 2018

2. *Class Diagram*

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam suatu sistem. *Class Diagram* memberikan sebuah gambaran statis tentang sistem beserta relasi-relasi yang ada di dalam sistem tersebut. (Novianti & Fauzijah, 2009).

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. (Agung et al., 2018).

Menurut Agung et al (2018) *Class Diagram* menggambarkan keadaan diantaranya :

- a. *Atribut/properti* suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi),
- b. Menggambarkan struktur dan deskripsi class, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Menurut Agung et al (2018), *Class Diagram* memiliki tiga area pokok :


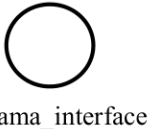





- a. Nama (*stereotype*),
- b. Atribut,
- c. Metoda.

Menurut Agung et al (2018), atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan,
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya,
- c. *Public*, dapat dipanggil oleh siapa saja.

Berikut adalah simbol-simbol yang ada pada *ClassDiagram* menurut Heriyanto (2018) :

Tabel 2.2 Simbol pada *Class Diagram*

SIMBOL	DESKRIPSI
Kelas 	Kelas pada struktur sistem.
Antarmuka/ <i>interface</i> 	Sama dengan <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi ini biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

Sumber : Yunahar Heriyanto, 2018

3. *Sequence Diagram*




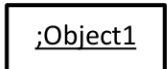
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem termasuk *user*, *display*, dan sebagainya yang berupa *message-message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas garis vertikal berupa waktu dan garis horizontal berupa objek-objek yang terkait. *Sequence Diagram* biasanya digunakan untuk menggambarkan skenario atau rangkaian-rangkaian yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. (Novianti & Fauzijah, 2009)


Menurut Novianti & Fauziah (2009), dalam sistem akses atau ubah data, *Sequence Diagram* dibagi berdasar pengelompokkan *Class Diagram*. Setiap proses dalam *Sequence Diagram* ini memiliki interaksi antar objek yang berurutan. Dalam penggambaran *Sequence Diagram* terdapat beberapa objek yang saling berinteraksi, yaitu:

- a. *Actor:User* yang bersangkutan,
- b. *Class Boundary:Class* tampilan sistem,
- c. *Class Control:Class* yang merupakan penghubung antara *class* tampilan dan *class* entitas,
- d. *Class Entity:Class* entitas dari sistem (Objek).

Berikut adalah simbol-simbol yang ada pada *Sequence Diagram* menurut Heriyanto (2018) :

Tabel 2.3 Simbol pada *Sequence Diagram*

NAMA KOMPONEN	KETERANGAN	SIMBOL
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	
<i>Message</i>	<i>Message</i> digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek-objek.	
<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama <i>object</i> didalamnya yang diawali dengan sebuah titik koma.	

NAMA KOMPONEN	KETERANGAN	SIMBOL
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi.	



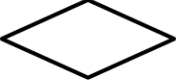

Sumber : Yunahar Heriyanto, 2018

4. *Activity Diagram*

Activity Diagram menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses, yang biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas bisnis. Selain bisnis, *Activity Diagram* juga dapat digunakan menggambarkan *logical procedural* sistem, dan aliran kerja kasus lainnya.

Berikut adalah simbol-simbol yang ada pada *Activity Diagram* menurut Heriyanto (2018) :

Tabel 2.4 Simbol pada *Activity Diagram*

SIMBOL	DESKRIPSI
Simbol <i>Start</i> 	Simbol <i>Start</i> untuk menyatakan awal dari suatu proses.
Simbol <i>Stop</i> 	Simbol <i>Stop</i> untuk menyatakan akhir dari suatu proses.
Simbol <i>Decision</i> 	Simbol <i>Decision</i> digunakan untuk menyatakan kondisi dari suatu proses.
Simbol <i>Action</i> 	Simbol <i>Action</i> menyatakan aksi yang dilakukan dalam suatu arsitektur sistem.

Sumber : Yunahar Heriyanto, 2018

J. *Sistem*

Berikut merupakan pengertian sistem menurut para ahli dan konsep dasar sistem :

1. Menurut Fat pengertian Sistem adalah suatu himpunan suatu benda nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan (*Unity*) untuk mencapai tujuan tertentu secara efisien dan efektif.
2. Pengertian Sistem Menurut Indrajit (2001: 2) mengemukakan bahwa sistem mengandung arti kumpulan-kumpulan dari komponen-komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya.
3. Pengertian Sistem Menurut Jogianto (2005: 2) mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi.
4. Pengertian Sistem Menurut Murdick, R.G, (1991: 27) Sistem adalah seperangkat elemen yang membentuk kumpulan atau *procedure-procedure*/bagan-bagan pengolahan yang mencari suatu tujuan tertentu.
5. Pengertian Sistem Menurut Jerry FutzGerald, (1981 : 5) Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan,

berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

6. Pengertian Sistem Menurut Davis, G.B, (1991 : 45) Sistem secara fisik adalah kumpulan dari elemen-elemen yang beroperasi bersama-sama untuk menyelesaikan suatu sasaran
7. Definisi Sistem Menurut Dr. Ir. Harijono Djojodihardjo (1984: 78) Sistem adalah sekumpulan objek yang mencakup hubungan fungsional antara tiap-tiap objek dan hubungan antara ciri tiap objek, dan yang secara keseluruhan merupakan suatu kesatuan secara fungsional.
8. Definisi Sistem Menurut Lani Sidharta (1995: 9) Sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara bersama mencapai tujuan-tujuan yang sama.

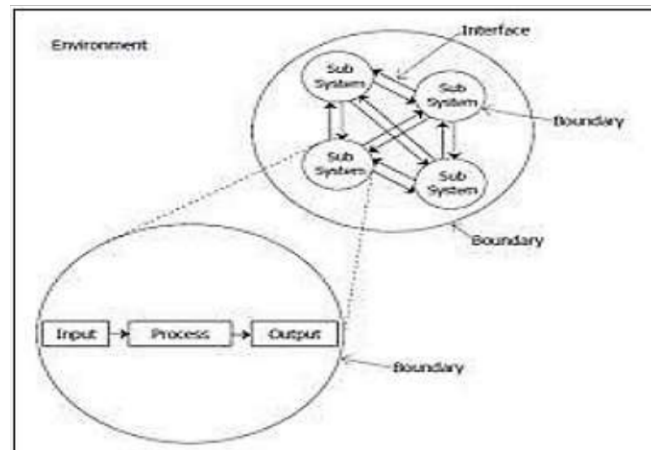
Dengan demikian sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Menurut Richard F. Neuschel prosedur adalah suatu urutan operasi klerikal (tulis-menulis), yang melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.(Hutahean, 2014).

1. Karakteristik Sistem

Menurut Hutahean (2014), supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu :

- a. Komponen Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.
- b. Batasan sistem (*boundary*). Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
- c. Lingkungan luar sistem (*environment*). Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.
- d. Penghubung sistem (*interface*). Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

- e. Masukkan Sistem (*input*). Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.
- f. Keluaran sistem (*output*). Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.
- g. Pengolah sistem Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, system akuntansi akan mengolah data menjadi laporan-laporan keuangan.
- h. Sasaran sistem Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.



Gambar 2.10 Karakteristik dari Suatu Sistem
Sumber : Jeperson Hutahean, 2014

2. Klasifikasi Sistem

Menurut Hutahean (2014), sistem dapat diklasifikasikan dalam beberapa sudut pandang :

a. Klasifikasi sistem sebagai :

1. Sistem abstrak (*abstract system*). Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik,
2. Sistem fisik (*physical system*). Sistem fisik adalah sistem yang ada secara fisik.

b. Sistem diklasifikasikan sebagai :

1. Sistem alamiah (*natural system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi,
2. Sistem buatan manusia (*human made system*). Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

c. Sistem diklasifikasikan sebagai :

1. Sistem tertentu (*deterministic system*). Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan,
2. Sistem tak tentu (*probabilistic system*). Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

d. Sistem diklasifikasikan sebagai :

1. Sistem tertutup (*close system*). Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*,
2. Sistem terbuka (*open system*). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima *input* dan *output* dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendali yang baik.

3. Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi

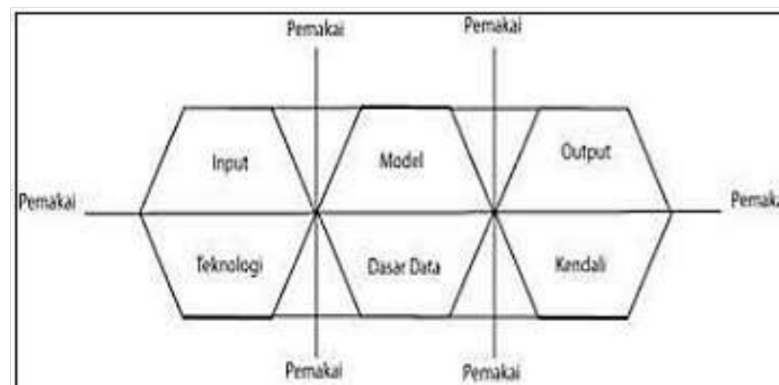
dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.(Hutahean, 2014)

4. Konsep Sistem Informasi

Menurut Hutahean (2014), sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

- a. Blok masukan (*input block*). *Input* mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.
- b. Blok model (*model block*). Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.
- c. Blok keluaran (*output block*). Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
- d. Blok teknologi (*technology block*). Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsur utama :

1. Teknisi (*human ware* atau *brain ware*),
 2. Perangkat lunak (*software*),
 3. Perangkat keras (*hardware*).
- e. Blok basis data (*database block*). Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
- f. Blok kendali (*control block*). Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.



Gambar 2.11 Blok Sistem Informasi yang Berinteraksi
 Sumber :Jeperson Hutahean, 2014

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kuantitatif yang bertujuan untuk mengetahui berapakah total penjualan *sparepart* dan penggunaan jasa servis setiap harinya dan mengetahui stok *sparepart* yang tersedia di Bengkel Suka Maju Mandiri Servis Dan Autopart. Berikut pengertian penelitian kuantitatif menurut para ahli :

1. Nana Sudjana dan Ibrahim (2001), Penelitian kuantitatif adalah penelitian yang didasari pada asumsi, kemudian ditentukan variabel, dan selanjutnya dianalisis dengan menggunakan metode-metode penelitian yang valid, terutama dalam penelitian kuantitatif.
2. Bryman (2005), Proses penelitian kuantitatif dimulai dari teori, hipotesis, desain penelitian, memilih subjek, mengumpulkan data, memproses data, menganalisa data, dan menuliskan kesimpulan.
3. Suriasumantri (2005), Penelitian kuantitatif adalah penelitian yang dilakukan dengan kajian pemikiran yang sifatnya ilmiah. Kajian ini menggunakan proses *logico-hypothetico-verifikatif* pada langkah-langkah penelitian yang dilakukan.
4. Kasiram (2008), Penelitian kuantitatif dapat didefinisikan sebagai suatu proses menemukan pengetahuan dengan menggunakan data berupa angka sebagai alat untuk menganalisis keterangan tentang apa yang ingin diketahui.

1. Tempat Penelitian

Penelitian ini dilakukan dengan survey langsung ke Bengkel Maju Mandiri Servis dan Autopart Pekanbaru untuk melakukan pengolahan data.

2. Alat Pengumpulan Data

Dalam pelaksanaan penelitian, dibutuhkan beberapa alat untuk membantu proses keberhasilan penelitian ini. Alat yang digunakan penelitian ini terdiri dari :

a. *Hardware*

1. Komputer,
2. *Keyboard dan Mouse*,
3. *Barcode Scanner*,
4. Printer.

b. *Software*

1. *Operating System* menggunakan *windows 10 pro*,
2. *Xampp*,
3. *MySQL* digunakan untuk *database*,
4. *PHP* digunakan untuk dasar pemrograman *web*,
5. *Google Chrome* digunakan untuk *web browser*,
6. *StarUML* digunakan untuk membuat diagram *UML*,
7. *Mandelely* digunakan untuk membuat sitasi.

B. Teknik Pengumpulan Data

1. Wawancara

Wawancara adalah percakapan yang dilakukan oleh dua belah pihak untuk saling bertukar informasi dengan cara tanya jawab dengan narasumber secara langsung maupun secara tidak langsung, sehingga mendapatkan sebuah kesimpulan dari wawancara tersebut.

2. Dokumen atau Studi Pustaka

Teknik pengumpulan data dengan mencari data dari jurnal, artikel, buku dan penelitian yang sudah ada, untuk menggali sebuah informasi yang dibutuhkan untuk penelitian.

C. Jenis Data

1. Data Primer

Berikut adalah pengertian data primer menurut para ahli :

- a. Menurut Sugiyono (2016: 225), data primer merupakan sumber data yang langsung memberikan data kepada pengumpul data. Sumber data primer didapatkan melalui kegiatan wawancara dengan subjek penelitian dan dengan observasi atau pengamatan langsung di lapangan.
- b. Menurut Husein Umar (2013:42), data primer merupakan data yang didapat dari sumber pertama baik dari individu atau perorangan seperti hasil dari wawancara atau hasil pengisian kuesioner yang biasa dilakukan oleh peneliti.

- c. Menurut Nur Indrianto dan Bambang Supono (2013:142), data primer merupakan sumber data penelitian yang diperoleh langsung dari sumber asli (tidak melalui media perantara).
- d. Menurut Kuncoro (2009 : 145), data primer adalah data yang didapat dan dikumpulkan dari sumber-sumber asli untuk tujuan tertentu.
- e. Menurut Sanusi (2012), data primer adalah data yang pertama kali dicatat dan dikumpulkan oleh peneliti.
- f. Menurut Danang Sunyoto (2013:21), data primer adalah data asli yang dikumpulkan sendiri oleh peneliti untuk menjawab masalah penelitiannya secara khusus.
- g. Menurut Arikunto (2013:22) Data primer adalah data dalam bentuk verbal atau kata-kata yang diucapkan secara lisan, gerak gerik atau perilaku yang dilakukan oleh subjek yang dapat dipercaya, dalam hal ini adalah subjek penelitian (informan) yang berkenaan dengan variabel yang diteliti.

Berikut adalah data *sparepart* Bengkel Maju Mandiri Servis dan Autopart yang didapat peneliti kepada narasumber melalui wawancara :

Tabel 3.1 Data Sparepart Bengkel Maju Mandiri Servis dan Autopart

Nama	Model	Stok	Harga
Jumbo gear Oil 1 Liter	-	22	40.000
Oli United XTG-9 1 Liter	-	2	50.000
Unioil 1 Liter	-	4	45.000
Gear Oil Top1 1 liter	-	26	65.000
Oli Hp Sport Top1 1 Liter	-	5	70.000
Oli ATF MV Top1 1 Liter	-	4	45.000
Oli Multi Power Motul 1 Liter	-	5	120.000
Radiator Coolant Prestise 5 Liter	-	10	55.000
Radiator Treatment Wealthy 5 Liter	-	25	120.000

Nama	Model	Stok	Harga
-------------	--------------	-------------	--------------

Radiator Coolant Jumbo 5 Liter	-	3	80.000
Oli Total 3000 1 Liter	-	7	65.000
Oli Total 5000 1 Liter	-	3	75.000
Oli Total 5000 4 Liter	-	4	300.000
Oli Total 7000 1 Liter	-	5	90.000
Oli Total Rubia 1 Liter	-	4	75.000
Oli Total Transtec 4 1 Liter	-	8	85.000
Water Pump GNS	Avanza/DI Xenia/Grandmax	7	235.000
Hydraulic Sanyco	Grandmax	4	225.000
Hydraulic Parts Birkens	Kijang 7k	6	425.000
Brake Master CYL-ASSY	Avanza	6	425.000
Wheel Hub, Front	L300	7	335.000
Clutch Master CYL-ASSY	Kijang 7K	4	225.000
Master Cylinder	Grandmax	3	175.000
Engine Mounting	Terios	4	195.000
Tank Unit	Avanza	4	150.000
In-Tank Fuel Pump	Inova	3	225.000
In-Tank Fuel Pump	Avanza	6	225.000
In-Tank Fuel Pump	Grandmax	5	225.000
Fuel Pump Denso	Kijang 7K	8	450.000
Fuel Pump Daihatsu	Zebra	9	275.000
Bearing Tapper Roller	Inova	11	180.000
Clutch Operating	Kijang 7K	4	175.000
Clutch Master	Terios	3	225.000
Wiper	Welty	7	250.000
Wiper	Akio	9	90.000
Wiper	Sakura	12	90.000
Belting	Avanza	8	120.000
Belting	Terios	6	135.000
Belting	Grandmax	7	95.000
Belting	Inova	10	250.000
Belting	APV	6	120.000
Tali Kopleng	Avanza	6	250.000
Tali Kopleng	Grandmax	10	220.000
Tali Kopleng	Apv	9	220.000
Karbu Cleaner Kecil	-	10	20.000
Karbu Cleaner Besar	-	12	45.000
Trotle Cleaner	-	10	95.000
Fuel Conditioner	-	10	75.000
Minyak Rem Jumbo Besar	-	15	60.000
Minyak Rem Jumbo Kecil	-	17	25.000
Minyak Rem Prestis Besar	-	13	60.000
Minyak Rem Prestis Kecil	-	18	25.000
Gomok Deltalub	-	7	85.000
Gomok Cobra	-	9	30.000
Gomok Timac	-	10	40.000
Rotary Syntetic Greas	-	6	50.000
Wheel Bearing Greas	-	8	45.000
WD 40	-	7	35.000

2. Data Sekunder

Berikut adalah pengertian data sekunder menurut para ahli :

- a. Menurut Sugiyono (2016: 225), data sekunder merupakan sumber data yang tidak langsung memberikan data kepada pengumpul data, misalnya melalui orang lain atau lewat dokumen. Sumber data sekunder merupakan sumber data pelengkap yang berfungsi melengkapi data yang diperlukan data primer.
- b. Menurut Hasan (2002: 58), data sekunder adalah data yang diperoleh atau dikumpulkan oleh orang yang melakukan penelitian dari sumber-sumber yang telah ada. Data ini digunakan untuk mendukung informasi primer yang telah diperoleh yaitu dari bahan pustaka, literatur, penelitian terdahulu, buku, dan lain sebagainya.
- c. Menurut Husein Umar (2013:42), data sekunder merupakan data primer yang telah diolah lebih lanjut dan disajikan baik oleh pihak pengumpul data primer atau oleh pihak lain misalnya dalam bentuk tabel-tabel atau diagram- diagram.
- d. Menurut Nur Indrianto dan Bambang Supomo (2013:143), data sekunder merupakan sumber data penelitian yang diperoleh peneliti secara tidak langsung melalui media perantara (diperoleh dan dicatat oleh pihak lain).
- e. Menurut Arikunto (2013:22), data sekunder adalah data yang diperoleh dari dokumen-dokumen grafis (tabel, catatan, notulen rapat, SMS, dan lain-lain), foto-foto, film, rekaman video, benda-benda dan

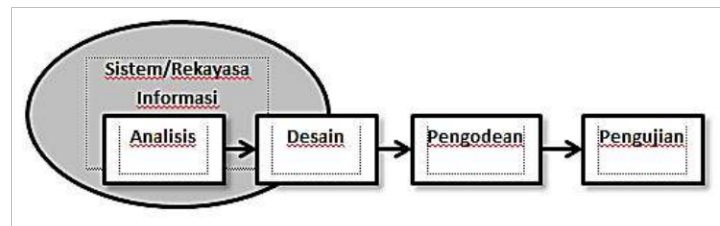
lain-lain yang dapat memperkaya data primer dapat memperkaya data primer

- f. Menurut Kuncoro (2009:145), data sekunder adalah data yang telah dikumpulkan oleh pihak lain, peneliti dapat mencari sumber data ini melalui sumber data lain yang berkaitan dengan data yang ingin dicari.
- g. Menurut Sanusi (2012), data sekunder adalah data yang sudah tersedia dan dikumpulkan oleh pihak lain di luar instansi yang diteliti.

D. Teknik Pengembangan Sistem

1. Pengembangan Model *Waterfall*

Menurut Rosa A.S. dan M. Shalahuddin (2014:31) dalam Hidayat et al (2017), *Waterfall* adalah model *SDLC* paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah. Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequentiallinear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun menurut Rosa A.S. dan M. Shalahuddin (2014:29) dalam Hidayat et al (2017), yaitu:



Gambar 3.1 Model Waterfall
Sumber : Hidayat et al, 2017

- a. Analisis kebutuhan perangkat lunak Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
- b. Desain Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.
- c. Pembuatan kode program Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
- d. Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

- e. Pendukung (*support*) atau pemeliharaan (*maintenance*) tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

BAB IV

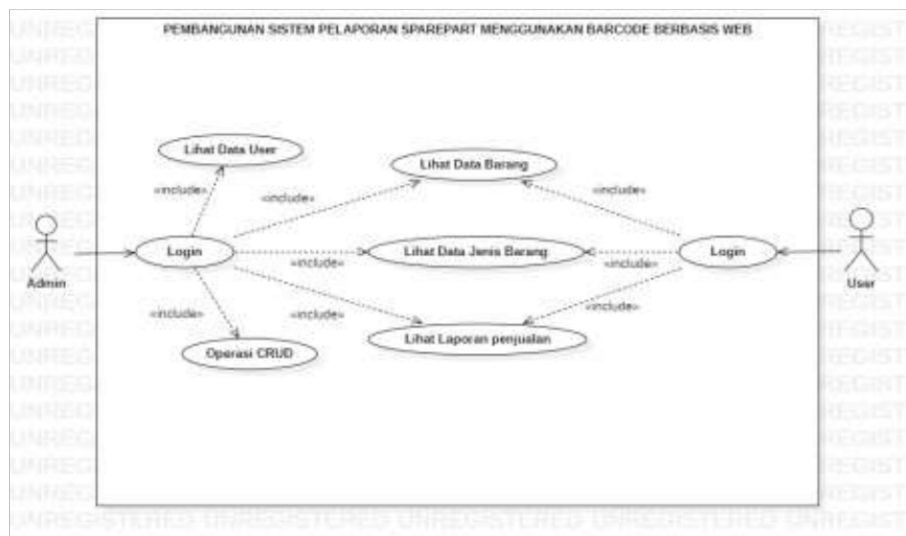
Hasil dan Pembahasan

A. Pembahasan

Pembahasan ini akan membahas bagaimana sistem ini bekerja sehingga dapat mempermudah admin dalam melakukan pelaporan. Sistem ini bekerja dengan cara menginput data kedalam *Database* menggunakan sistem yang dibuat yaitu Sistem Pelaporan *Sparepart* Menggunakan *Barcode* Berbasis Web. Admin dapat melakukan *Input* dan *Output* data barang ke *Database* dengan mudah, sehingga akan mendapatkan suatu data stok barang yang *valid*. Data yang di *input* ke *database* merupakan data *valid* hasil dari wawancara dengan narasumber secara langsung, sehingga *user* bisa melakukan pendataan secara *manual* ke lokasi studi kasus yaitu Bengkel Maju Mandiri Servis Dan Autopart.

B. Use Case Diagram

Use case diagram menggambarkan interaksi antara admin dan *user* dengan sistem dalam Pembangunan Sistem Pelaporan Sparepart Menggunakan Barcode Berbasis Web Studi Kasus Bengkel Maju Mandiri Servis Dan Autopart.



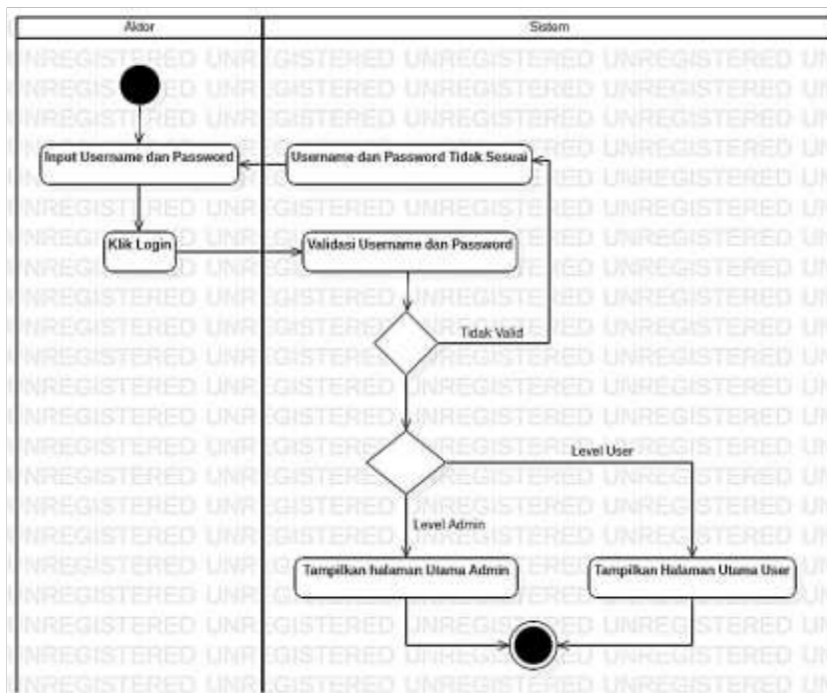
Gambar 4.1 Use Case Diagram Pembangunan Sistem Pelaporan Sparepart

Use Case Diagram di atas menjelaskan admin atau *user* harus melakukan *Login* terlebih dahulu, lalu *user* hanya dapat mengelola transaksi penjualan, lihat data barang, cetak *Barcode*, lihat laporan penjualan. Sedangkan admin dapat melakukan tambah data barang, *user* dan jenis barang, lalu melakukan edit dan hapus data.

C. Activity Diagram

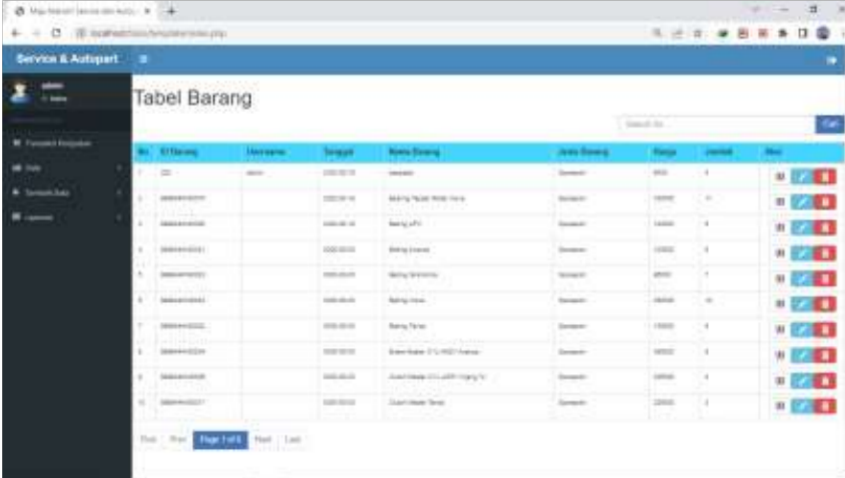
Activity Diagram menggambarkan suatu aktifitas yang terjadi antara admin atau *user* dengan sistem dalam Sistem Pelaporan Sparepart Menggunakan *Barcode* Berbasis Web.

1. Activity Diagram Login



Gambar 4.2 Activity Diagram Login

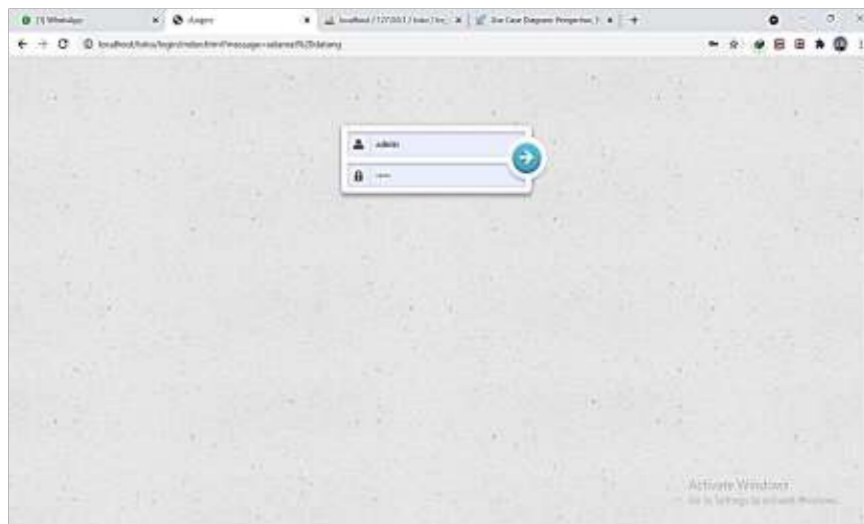
Activity Diagram di atas menjelaskan kegiatan Aktor yaitu admin atau *user* dalam melakukan *login* agar dapat mengakses kedalam sistem yang dibuat, dengan memasukkan *username* dan *password* lalu *user* mengklik tombol *login*. Sistem akan melakukan validasi pada *username* dan *password* yang dimasukkan. Jika *username* dan *password* benar maka sistem akan melakukan validasi level berdasarkan *username* tersebut lalu menampilkan halaman utama, dan jika *username* dan *password* salah maka sistem akan menampilkan kembali halaman *login*.



The screenshot shows a web browser window with the URL 'http://localhost:3000/produk/index.php'. The page title is 'Service & Autogart'. The main content is a table titled 'Tabel Barang'. The table has the following columns: No., ID Barang, Nama Barang, Stok, Status, Harga, Jumlah, and Aksi. The table contains 10 rows of data.

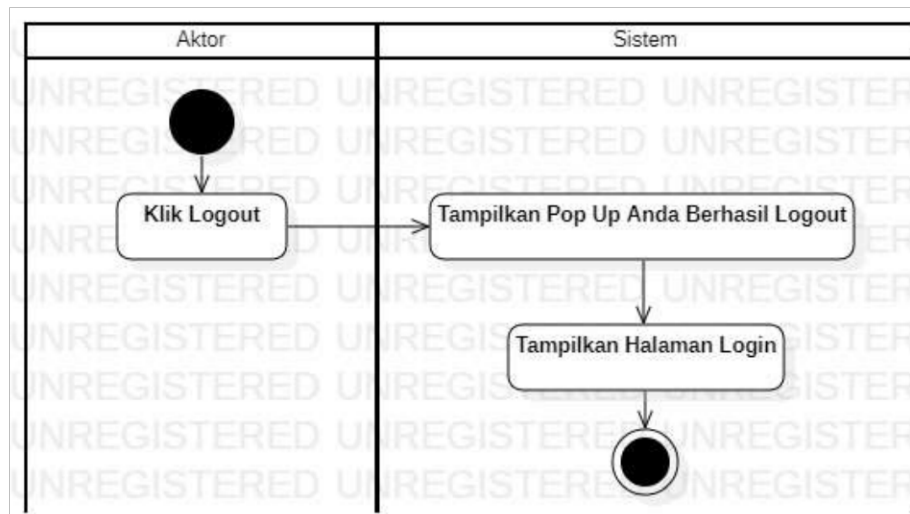
No.	ID Barang	Nama Barang	Stok	Status	Harga	Jumlah	Aksi
1	0000000001	Minyak Motor 4000 cc	10000	Stok	10000	1	[Edit] [Delete]
2	0000000002	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
3	0000000003	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
4	0000000004	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
5	0000000005	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
6	0000000006	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
7	0000000007	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
8	0000000008	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
9	0000000009	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]
10	0000000010	Minyak Oli	10000	Stok	10000	1	[Edit] [Delete]

Gambar 4.3 Tampilan Jika Berhasil Login



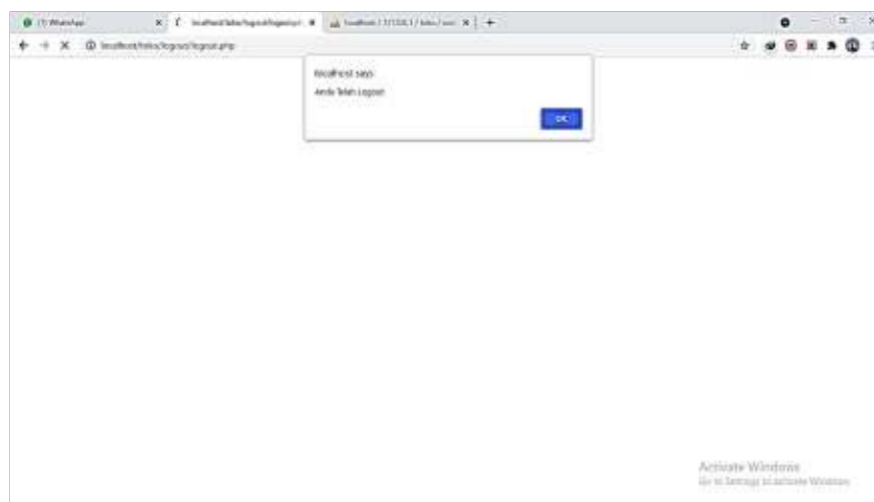
Gambar 4.4 Tampilan jika Gagal Login

2. Activity Diagram Logout



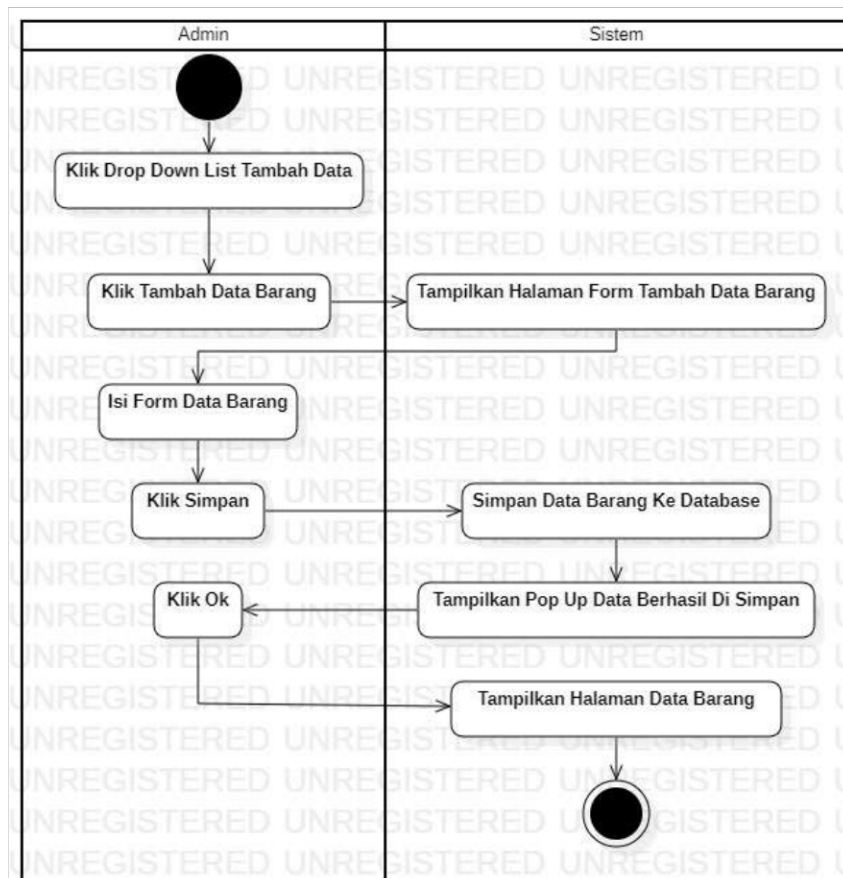
Gambar 4.5 Activity Diagram Logout

Activity Diagram diatas menjelaskan kegiatan Aktor saat melakukan *logout* dari sistem pelaporan *sparepart* menggunakan *barcode* berbasis web. Aktor mengklik tombol *logout* pada halaman website, lalu sistem akan tampilkan *Pop Up* dan membuka halaman *login*.



Gambar 4.6 Tampilan Berhasil Logout

3. Activity Diagram Tambah Data Barang

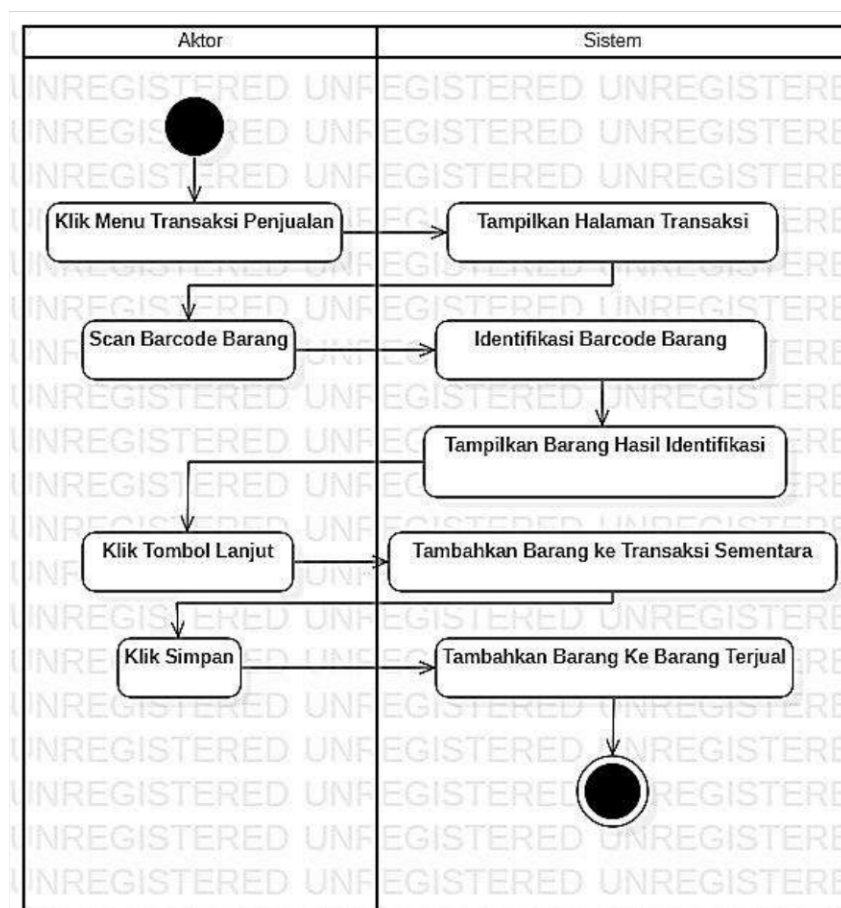


Gambar 4.7 Activity Diagram Tambah Data Barang

Activity Diagram di atas menunjukkan aktifitas yang dilakukan admin saat ingin tambah data barang. Admin mengklik *drop down list* tambah data, lalu klik tambah data barang pada *list* tersebut, lalu sistem akan menampilkan *form* yang perlu diisi untuk menambahkan data barang. Admin mengklik *button* simpan saat selesai mengisi *form*, lalu sistem akan menyimpan data tersebut dan menampilkan *Pop Up* Data Berhasil Disimpan, setelah selesai sistem akan tampilkan halaman data barang.

Gambar 4.8 Form Tambah Data Barang

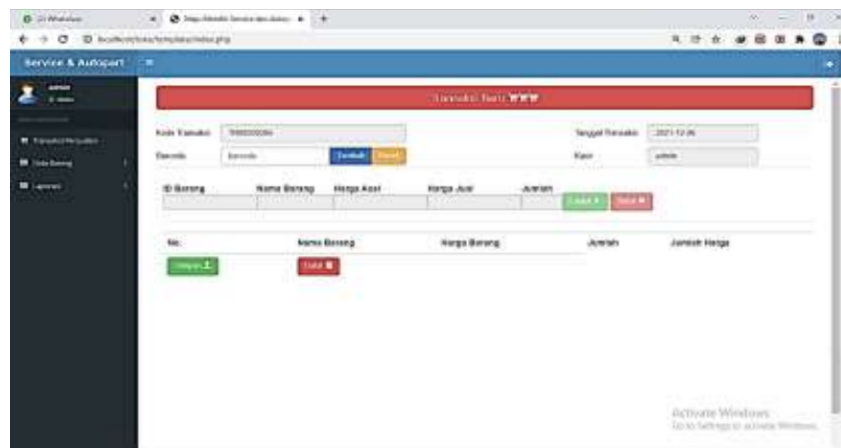
4. Activity Transaksi Penjualan



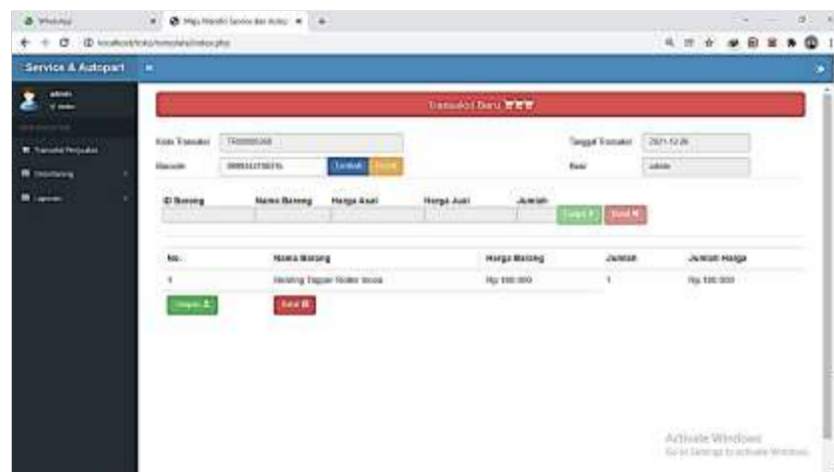
Gambar 4.9 Activity Transaksi Penjualan

Gambar di atas merupakan aktifitas diagram yang menunjukkan aktifitas yang dilakukan Aktor pada saat melakukan penjualan *sparepart*

guna untuk melakukan *update* stok *sparepart* yang tersedia. Aktor melakukan *scan barcode* pada barang yang ingin dijual, lalu sistem akan mengidentifikasi *barcode* tersebut dan menampilkan data yang sesuai dari hasil identifikasi. Setelah itu, aktor klik tombol lanjut dan data akan ditambahkan ke transaksi sementara, guna untuk menambahkan penjualan *sparepart* yang lain, lalu aktor klik tombol simpan dan sistem akan menambahkan hasil jual tadi ke data barang terjual.

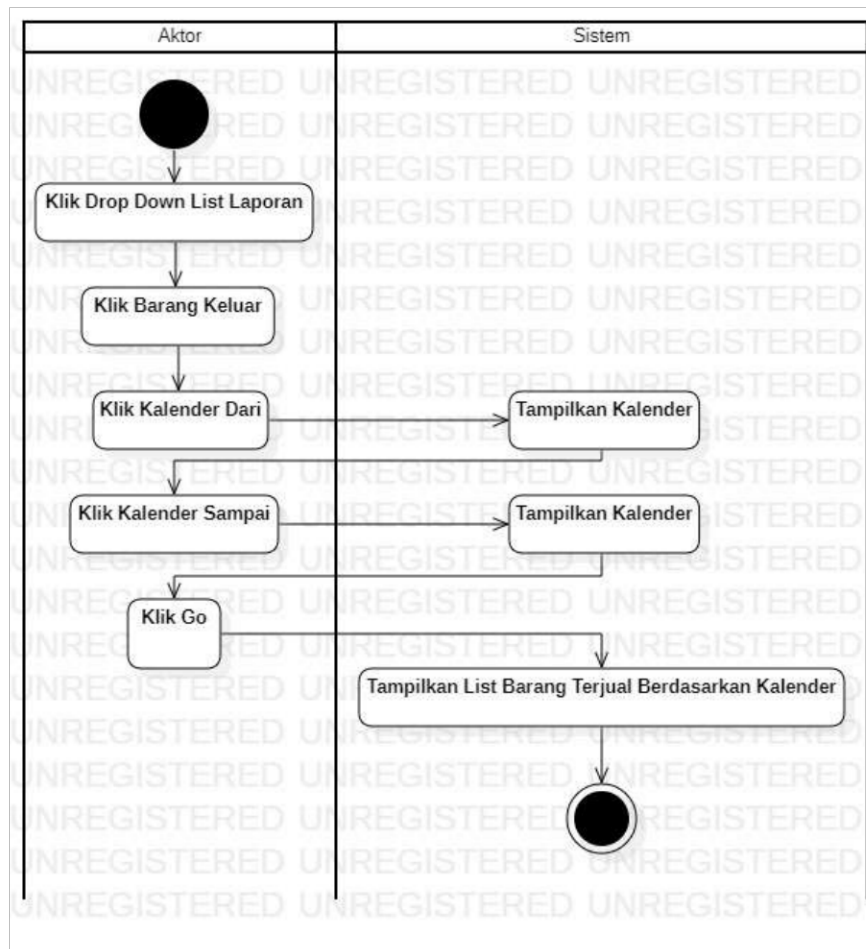


Gambar 4.10 Hasil Identifikasi *Barcode*



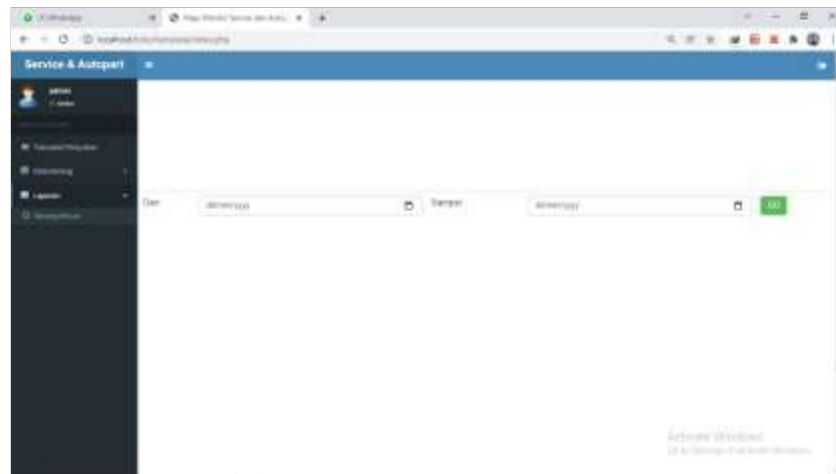
Gambar 4.11 Transaksi Sementara

5. Activity Diagram Barang Terjual



Gambar 4.12 Activity Diagram Barang Terjual

Gambar di atas merupakan aktifitas aktor saat ingin melihat laporan barang terjual sesuai dengan tanggal yang diatur sendiri oleh aktor, lalu sistem akan menampilkan *list* barang yang terjual sesuai dengan yang diatur, di dalam *list* ini juga terdapat total hasil penjualan sesuai dengan pengaturan kalender.

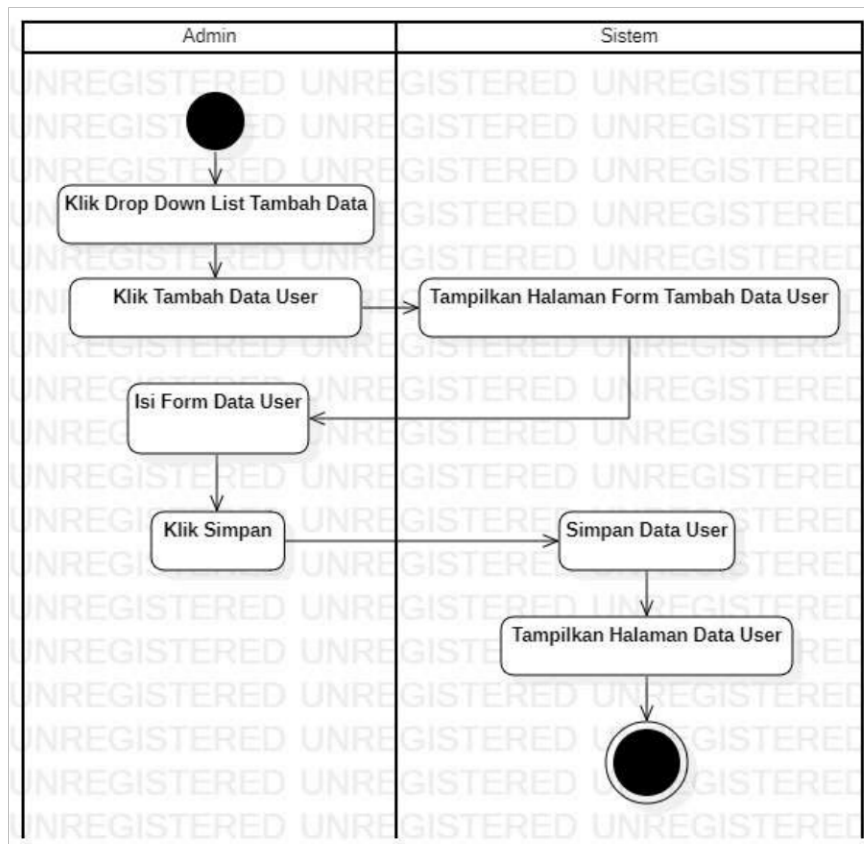


Gambar 4.13 Pengaturan Kalender

No.	Kode Transaksi	Kelas	ID Barang	Nama Barang	Jumlah	Harga	Harga Total	Tanggal
1	7100000027	apmm	002644150113	Bearing Tapper Roller Servis	1	Rp 180.000	Rp 180.000	2021-09-12
2	7100000028	apmm	002644150113	Bearing Tapper Roller Servis	1	Rp 180.000	Rp 180.000	2021-09-12
3	7100000029	apmm	002644150113	Bearing Tapper Roller Servis	1	Rp 180.000	Rp 180.000	2021-09-12
4	7100000078	apmm	0008000010108	kapak	1	Rp 8.000	Rp 8.000	2021-09-13
5	7100000038	apmm	0008000010108	kapak	1	Rp 8.000	Rp 8.000	2021-09-13
6	7100000038	apmm	0008000010108	kapak	1	Rp 8.000	Rp 8.000	2021-09-13
7	7100000038	apmm	0008000010108	kapak	1	Rp 8.000	Rp 8.000	2021-09-13
8	7100000002	apmm	002644150113	Bearing Tapper Roller Servis	1	Rp 180.000	Rp 180.000	2021-09-13
9	7100000027	apmm	002644150113	Bearing Tapper Roller Servis	1	Rp 180.000	Rp 180.000	2021-09-13
Total							Rp.824.000	

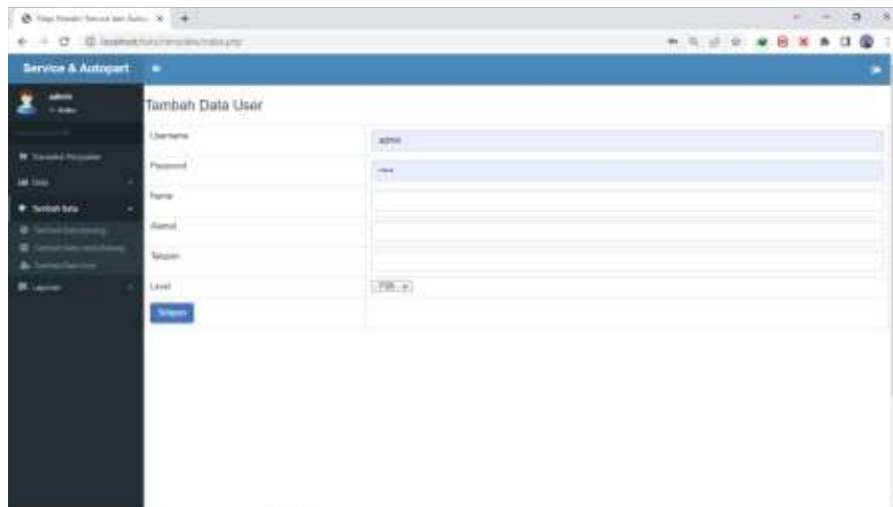
Gambar 4.14 List Barang Terjual Sesuai Pengaturan Kalender

6. *Activity Tambah Data User*



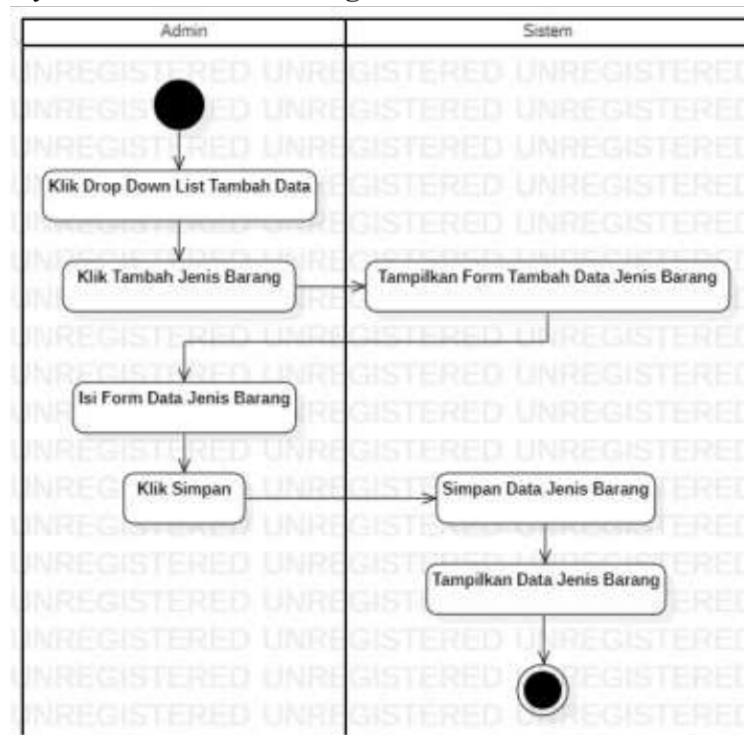
Gambar 4.15 *Activity Tambah Data User*

Aktifitas diatas merupakan tambah data pengguna yang hanya bisa dilakukan oleh admin. Aktifitas ini berfungsi untuk menambahkan *user* baru agar bisa melakukan *login*. Admin klik *drop down list* tambah data dan klik tambah data *user*, lalu sistem akan menampilkan *form* data user, admin melakukan pengisian data lalu klik simpan, kemudian sistam akan menyimpan data terserbut dan menampilkan kembali halaman data *user*.



Gambar 4.16 Halaman Form Data *User*

7. Activity Tambah Jenis Barang



Gambar 4.17 Activity Tambah Jenis Barang

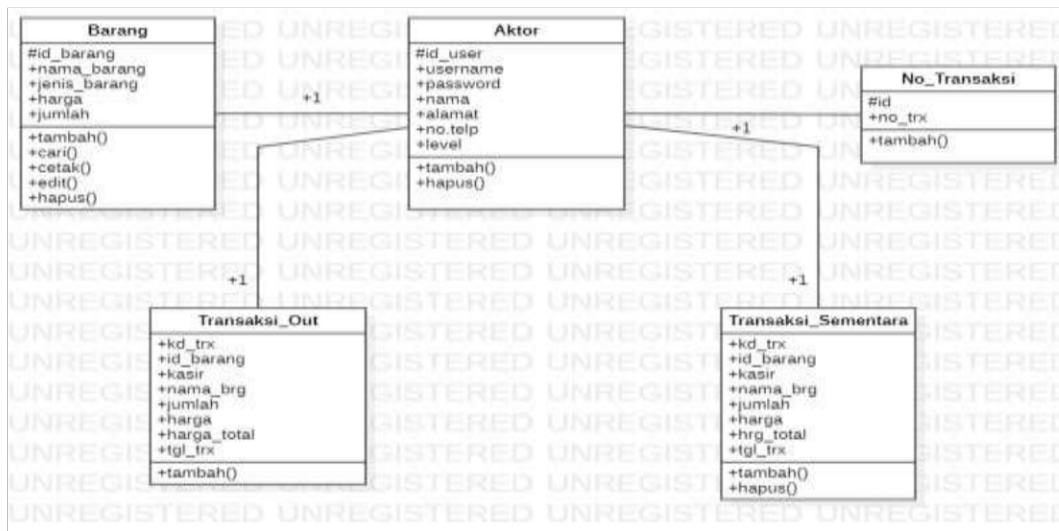
Aktifitas tambah jenis barang ini adalah tahap awal sebelum melakukan tambah data barang, admin klik *drop down list* tambah data dan klik tambah data jenis barang, lalu sistem akan menampilkan *form*

jenis barang, lalu admin isi *form* data jenis barang dan simpan, sistem akan simpan data jenis barang dan menampilkan halaman data jenis barang.

D. Class Diagram

Class Diagram merupakan hubungan *class* yang terdapat dalam sistem pelaporan *sparepart* yang digunakan untuk melihat gambaran suatu sistem beserta relasi di dalam sistem.

Dibawah ini merupakan gambar yang menunjukkan relasi didalam sistem pelaporan *sparepart*.

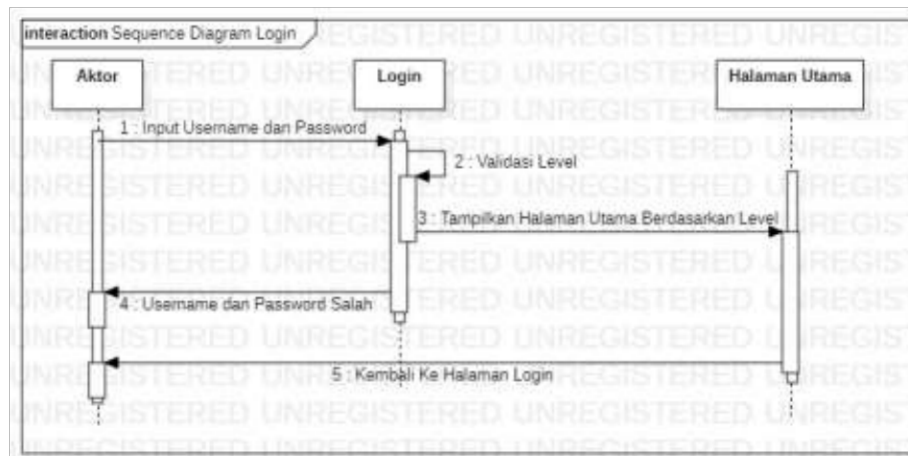


Gambar 4.18 *Class Diagram* Sistem Pelaporan *Sparepart*

E. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara objek berupa *message* yang digambarkan beserta waktu pelaksanaannya.

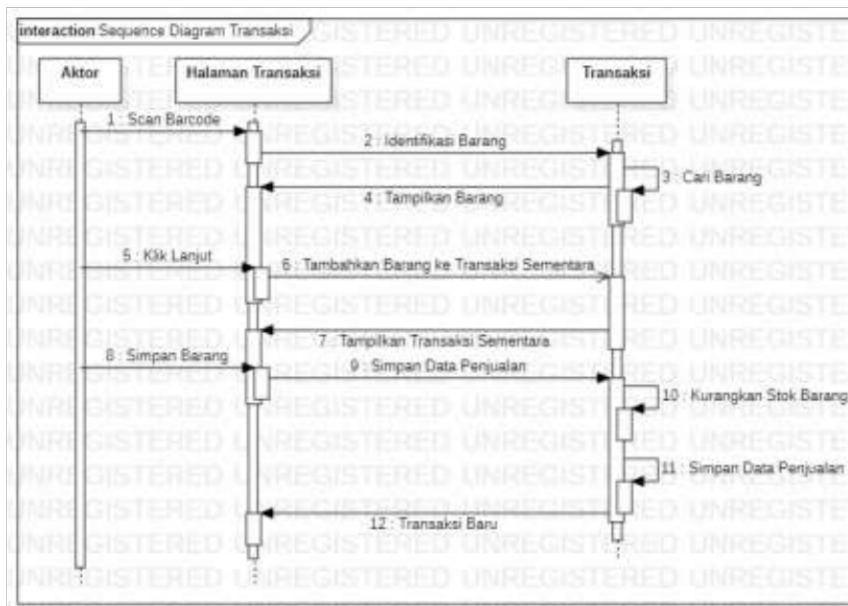
1. Sequence Diagram Login



Gambar 4.19 Sequence Diagram Login

Sequence diagram diatas menunjukkan bagaimana tahap saat melakukan proses *login*. Aktor memasukkan *username* dan *password*, lalu sistem akan melakukan verifikasi *username* dan *password*. Jika benar maka sistem akan melakukan validasi level berdasarkan *username*, lalu sistem akan menampilkan halaman utama berdasarkan level dari *username*, dan jika salah sistem akan mengembalikan aktor ke halaman *login*.

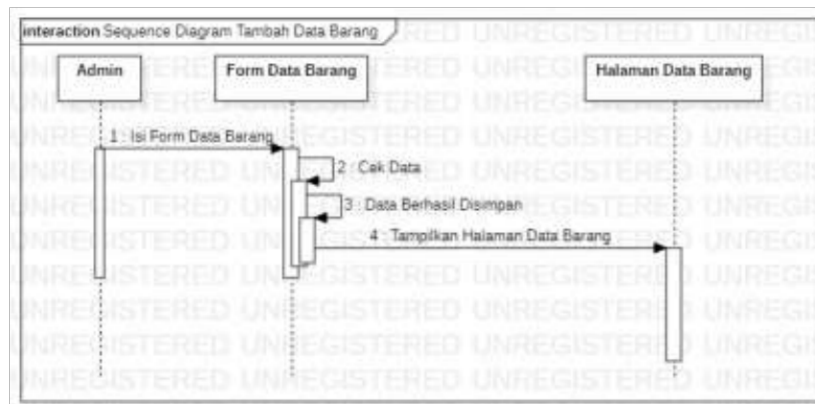
2. Sequence Diagram Transaksi Penjualan



Gambar 4.20 Sequence Diagram Transaksi Penjualan

Pada *sequence* diagram diatas menggambarkan urutan aktor dalam melakukan transaksi penjualan guna mengurangi stok barang. Aktor melakukan *scan barcode* pada *sparepart* yang sudah memiliki stiker *barcode*, lalu sistem transaksi akan mengidentifikasi dan mencari data *sparepart* sesuai dengan *code barcode*. Selanjutnya sistem akan menampilkan barang yang sesuai dengan *code barcode*, lalu aktor klik lanjut guna memasukkan data barang ke dalam transaksi sementara, dan sistem akan menampilkan data transaksi sementara. Aktor klik simpan barang lalu sistem akan mengkurangkan stok barang dan sistem akan menyimpan data penjualan, kemudian sistem akan menampilkan transaksi baru.

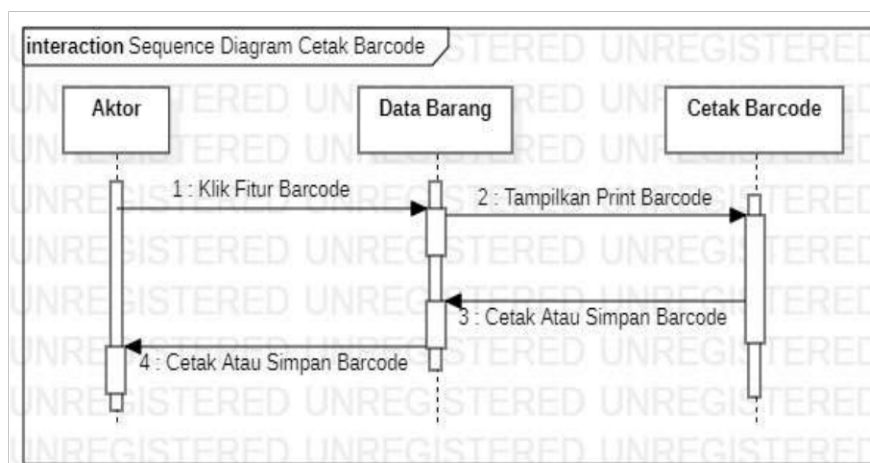
3. *Sequence Diagram* Tambah Data Barang



Gambar 4.21 *Sequence Diagram* Tambah Data Barang

Sequence diagram diatas menggambarkan bagaimana admin melakukan tambah data barang. Admin mengisi form data barang, setelah selesai sistem melakukan pengecekan *form*. Jika *form* sudah terisi semua maka sistem akan menyimpan data barang dan menampilkan *Pop Up* data berhasil disimpan, lalu sistem akan membuka halaman data barang.

4. *Sequence Diagram* Cetak Barcode

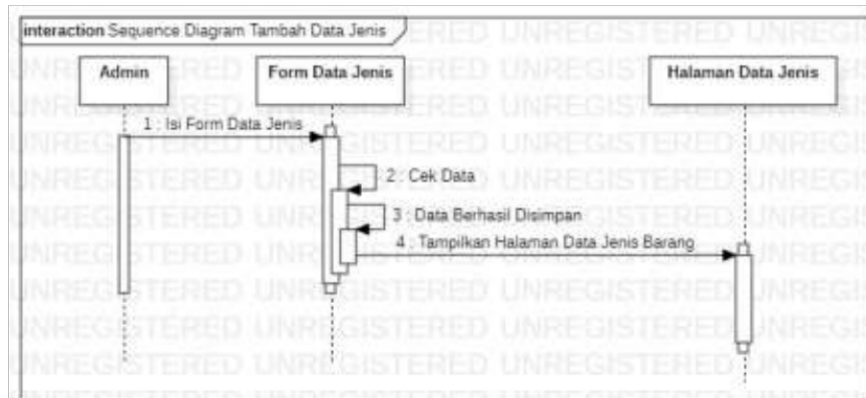


Gambar 4.22 *Sequence Diagram* Cetak Barcode

Pada gambar *sequence* diagram diatas menjelaskan bagaimana melakukan cetak atau simpan *code barcode* yang akan digunakan untuk mencetak stiker *barcode*. Aktor klik fitur *barcode* pada halaman data

barang, lalu sistem akan menampilkan *window print* berupa *code barcode*, lalu sistem akan mencetak atau menyimpan *code barcode* dan admin akan menerima *code barcode* tersebut.

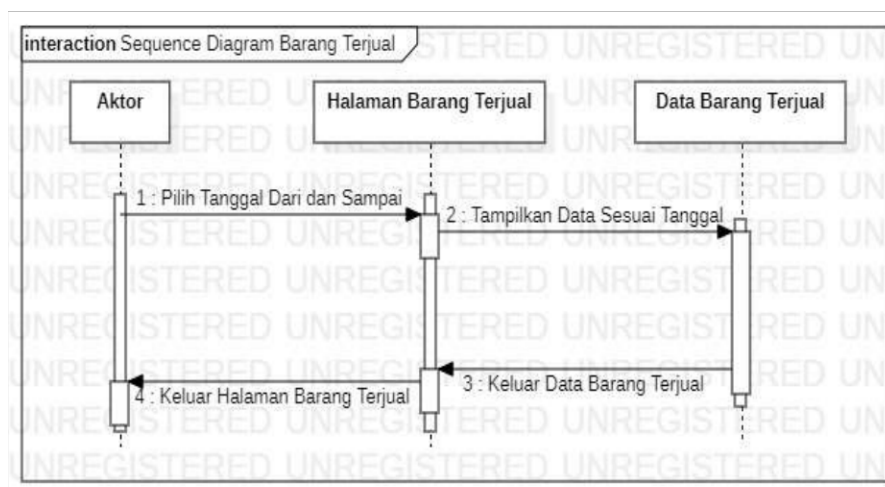
5. Sequence Diagram Tambah Data Jenis Barang



Gambar 4.23 Sequence Diagram Tambah Data Jenis Barang

Sequence diagram diatas menggambarkan bagaimana admin melakukan tambah data jenis barang, admin membuka tambah data jenis lalu mengisi form, lalu sistem akan mengecek data form dan menyimpan data tersebut, lalu sistem akan membuka halaman data jenis barang.

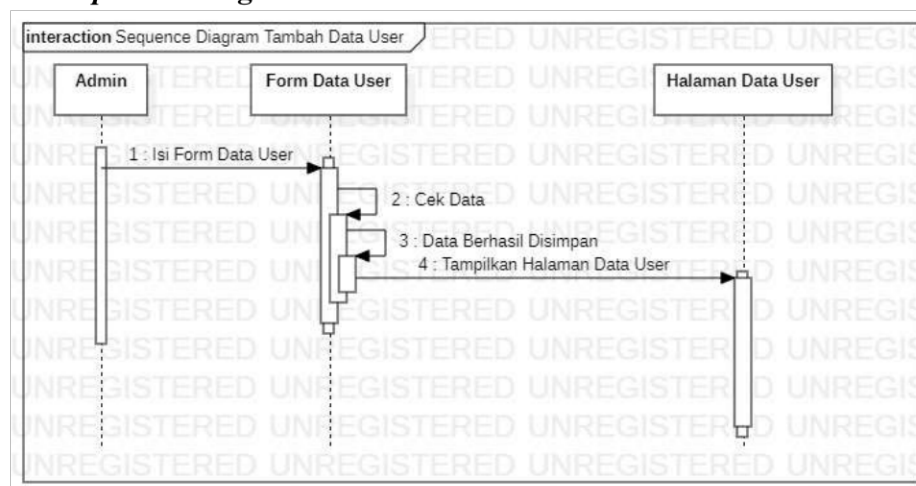
6. Sequence Diagram Barang Terjual



Gambar 4.24 Sequence Diagram Barang Terjual

Sequence diagram diatas menggambarkan bagaimana aktor dapat melihat data penjualan. Aktor akan memilih tanggal dari dan sampai, kemudian sistem akan menampilkan sistem data barang terjual sesuai dengan tanggal yang dipilih aktor, lalu aktor akan kembali ke halaman barang terjual setelah melihat laporan barang terjual.

7. *Sequence Diagram Tambah Data User*



Gambar 4.25 Sequence Diagram Tambah Data User

Sequence diagram diatas menggambarkan bagaimana admin melakukan penambahan data *user* baru, admin klik tambah data *user*, lalu sistem menampilkan halaman *form user*, lalu admin mengisi *form user* setelah klik simpan, sistem akan melakukan cek data lalu data berhasil disimpan, kemudian sistem akan tampilkan halaman data *user*.

F. *Database*

Database merupakan tempat penyimpanan data-data sistem yang saling berelasi. Berikut *database* yang digunakan dalam Pembangunan Sistem Pelaporan Menggunakan *Barcode* Berbasis Web.

1. Database User

Database ini digunakan untuk menyimpan data admin agar dapat melewati proses *login*.

Tabel 4.1 Database User

No	Nama	Type	Length	Keterangan
1	id_user	int	7	Primary Key
2	username	varchar	15	-
3	password	varchar	40	-
4	nama	varchar	20	-
5	alamat	text	-	-
6	no_tlp	text	-	-
7	level	varchar	10	-

2. Database Barang

Database ini merupakan tempat penyimpanan data-data barang sparepart yang tersedia di Bengkel Maju Mandiri Servis Dan Autopart.

Tabel 4.2 Database Barang

No	Nama	Type	Length	Keterangan
1	id_barang	varchar	15	Primary Key
2	kasir	varchar	15	-
3	tanggal	date	-	-
4	nama_brg	varchar	50	-
5	jenis_barang	varchar	30	-
6	harga	int	10	-
7	min	int	10	-
8	jumlah	int	5	-

3. Database Nomor Transaksi

Database ini digunakan untuk menyimpan data nomor transaksi yang nomor transaksinya dibuat secara otomatis oleh sistem.

Tabel 4.3 Database Nomor Transaksi

No	Nama	Type	Length	Keterangan
1	id	int	1	Primary Key
2	no_trx	int	8	-

4. Database Transaksi Sementara

Database ini merupakan tempat data penyimpanan transaksi sementara barang yang dibeli.

Tabel 4.4 Database Transaksi Sementara

No	Nama	Type	Length	Keterangan
1	kd_trx	varchar	10	-
2	id_brg	varchar	15	-
3	kasir	varchar	20	-
4	nama_brg	varchar	50	-
5	jumlah	int	3	-
6	harga	int	7	-
7	hrg_total	int	10	-
8	tgl_trx	date	-	-

5. Database Transaksi Penjualan

Database ini merupakan tempat penyimpanan data saat melakukan transaksi penjualan.

Tabel 4.5 Database Transaksi Penjualan

No	Nama	Type	Length	Keterangan
1	kd_trx	varchar	10	-
2	id_brg	varchar	15	-
3	kasir	varchar	20	-
4	nama_brg	varchar	50	-
5	jumlah	int	3	-
6	harga	int	10	-
7	hrg_total	int	10	-
8	tgl_trx	date	-	-

6. Database Batal Jual

Database ini digunakan jika pada transaksi sementara terjadi pembatalan beli *sparepart*, maka data pembatalan tersebut akan masuk ke *database* batal jual.

Tabel 4.6 Database Batal Jual

No	Nama	Type	Length	Keterangan
1	kd_trx	varchar	10	-
2	id_brg	varchar	15	-
3	kasir	varchar	20	-
4	nama_brg	varchar	50	-
5	jumlah	int	3	-
6	harga	int	10	-
7	hrg_total	int	10	-
8	tgl_trx	date	-	-

7. Database Jenis Barang

Database ini adalah tahap awal saat ingin melakukan tambah barang.

Tabel 4.7 Database Jenis Barang

No	Nama	Type	Length	Keterangan
1	id_jenis	int	10	Primary Key
2	jenis_barang	varchar	20	-

G. Implementasi Sistem

Implementasi sistem adalah prosedur yang dilakukan untuk menyelesaikan proses-proses yang dilakukan. Implementasi bertujuan untuk mengetahui apakah sistem sudah berjalan sesuai dengan yang diharapkan.

1. Halaman *Login*

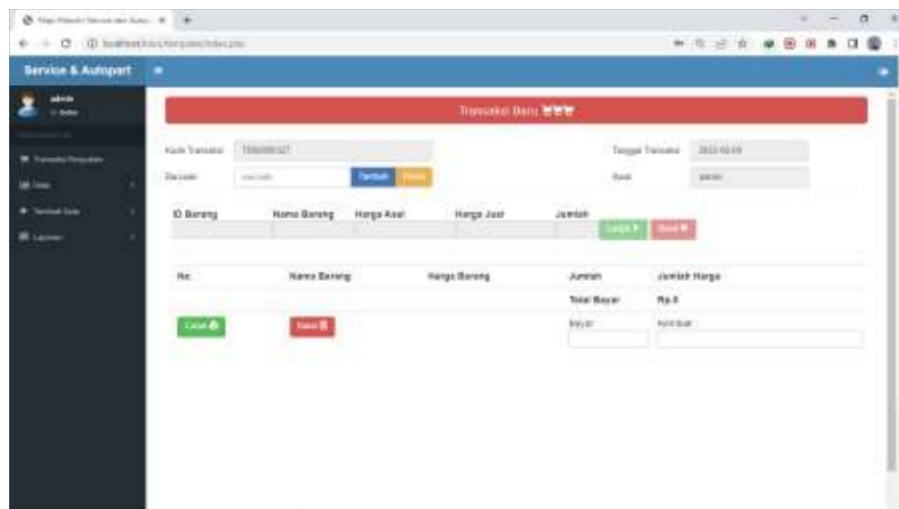
Login merupakan sebuah keamanan sistem agar sistem tidak dapat dibuka oleh sembarangan orang, jika *username* dan *password* benar maka sistem akan menampilkan halaman utama sistem.



Gambar 4.26 Halaman *Login* Sistem

2. Halaman Transaksi Penjualan

Halaman Transaksi Penjualan digunakan untuk admin dalam melakukan penjualan, sistem ini bekerja untuk menambahkan data pelaporan barang yang terjual dan mengupdate stok barang, sehingga data stok barang yang tersedia merupakan data yang *valid*.



Gambar 4.27 Halaman Transaksi Penjualan

3. Halaman Data Barang

Data Barang digunakan untuk menunjukkan halaman data barang yang sudah di *input* oleh admin dan untuk melihat barang yang tersedia beserta stok barang yang terdapat dalam Bengkel Maju Mandiri Servis Dan Autopart.

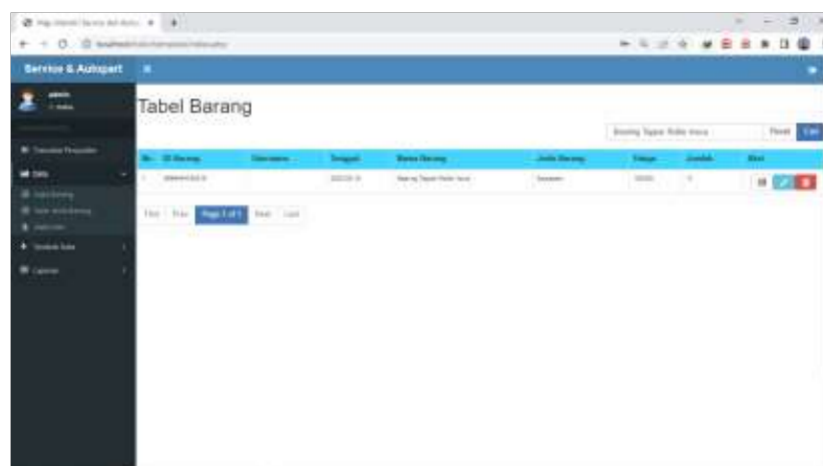
No.	ID Barang	Uraian	Tanggal	Nama Barang	Aksi Barang	Harga	Jumlah	Aksi
1	0000000001	amir	2022-02-01	amir	amir	10000	1	[edit] [delete]
2	0000000002		2022-02-01	barang mobil aksi mobil	amir	100000	1	[edit] [delete]
3	0000000003		2022-02-01	barang HP	amir	20000	1	[edit] [delete]
4	0000000004		2022-02-01	barang kacamata	amir	10000	1	[edit] [delete]
5	0000000005		2022-02-01	barang tas	amir	5000	1	[edit] [delete]
6	0000000006		2022-02-01	barang sepatu	amir	10000	1	[edit] [delete]
7	0000000007		2022-02-01	barang tas	amir	10000	1	[edit] [delete]
8	0000000008		2022-02-01	barang tas	amir	10000	1	[edit] [delete]
9	0000000009		2022-02-01	barang tas	amir	10000	1	[edit] [delete]
10	0000000010		2022-02-01	barang tas	amir	10000	1	[edit] [delete]

Gambar 4.28 Halaman Data Barang

Halaman ini menyediakan beberapa fitur yang diperlukan oleh admin untuk mengelola data yaitu :

a. Cari Barang

Fitur ini digunakan admin untuk mempermudah admin dalam melakukan pencarian barang.



Gambar 4.29 Fitur Cari Barang

b. Cetak *Barcode*

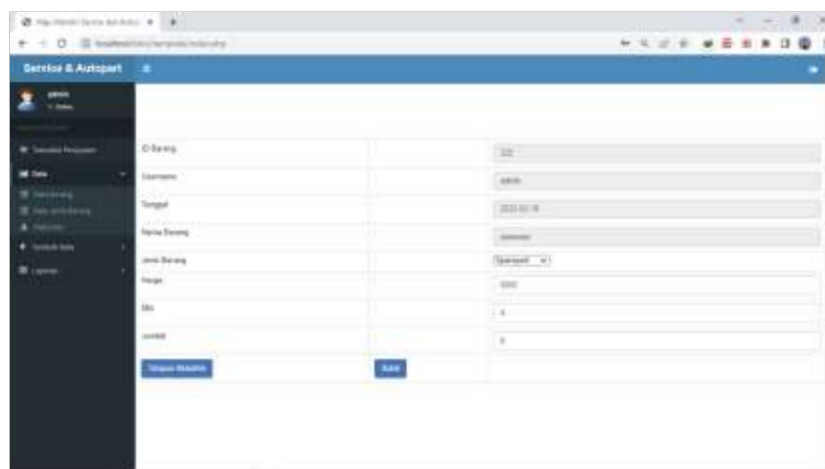
Fitur ini digunakan admin untuk melakukan cetak *barcode* barang jika ada barang yang belum memiliki stiker *barcode*.



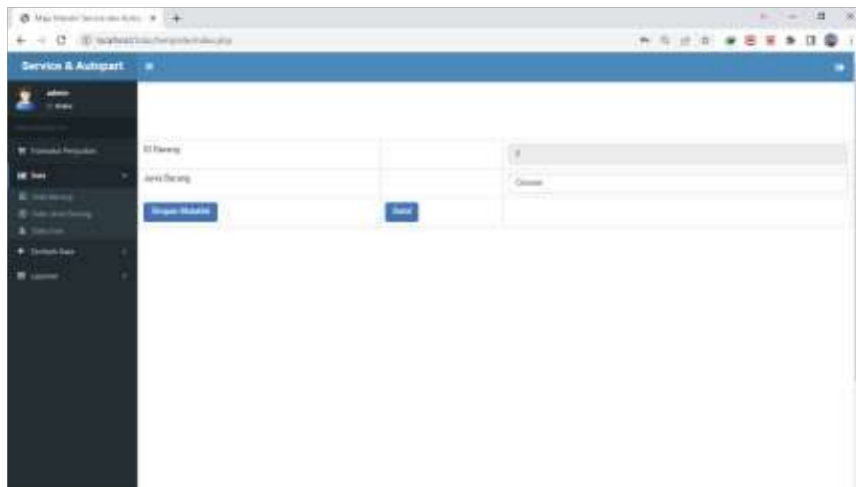
Gambar 4.30 Fitur Cetak *Barcode*

c. Edit Data Barang, Data Jenis dan Data *User*

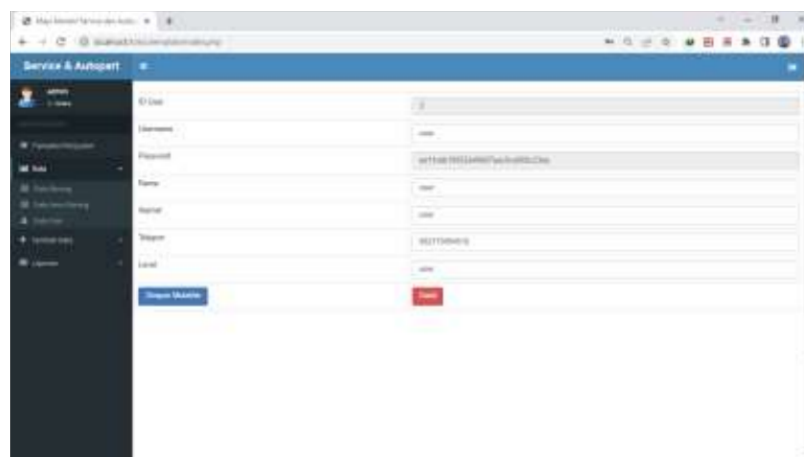
Fitur ini digunakan admin untuk melakukan *update* data barang jika ada penambahan stok barang serta melakukan perubahan harga barang jika ada barang yang harganya naik atau turun.



Gambar 4.31 Fitur Edit Barang



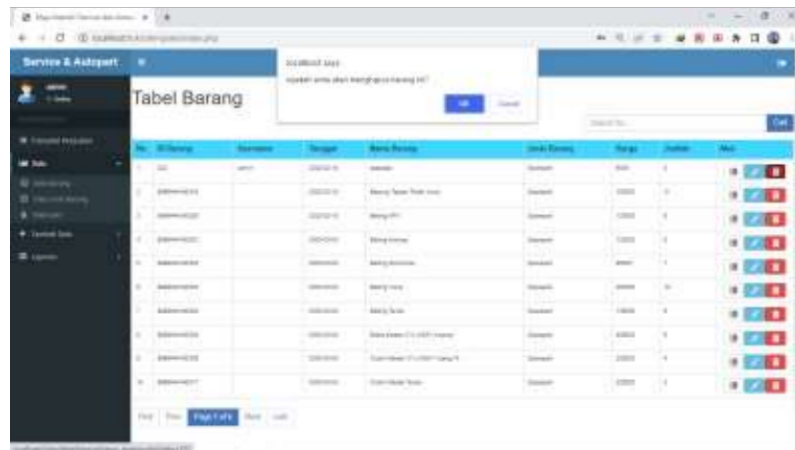
Gambar 4.32 Fitur Edit Jenis Barang



Gambar 4.33 Edit Data User

d. Hapus Barang

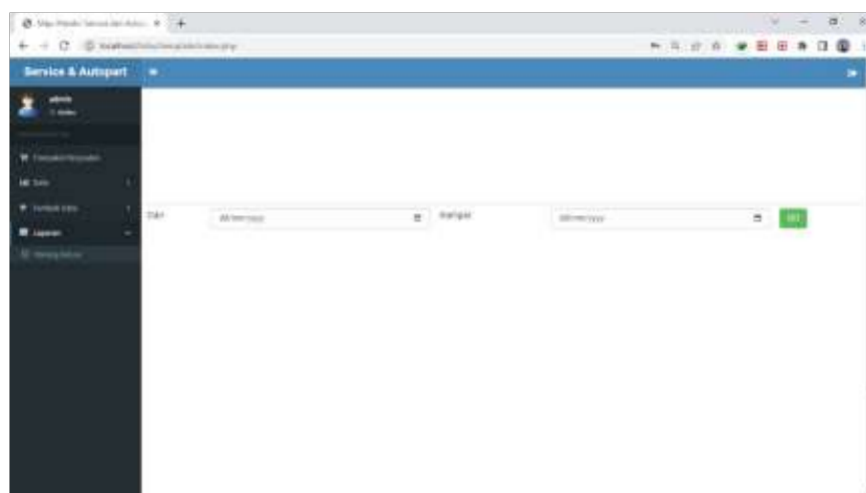
Fitur ini digunakan untuk menghapus data barang jika ada data barang yang tidak digunakan lagi, dikarenakan susah mencari stok barang yang tersedia.



Gambar 4.34 Fitur Hapus Barang

4. Halaman Barang Terjual

Menu Barang Terjual digunakan untuk menampilkan berapa total penjualan *sparepart*. Menu ini memiliki fitur kalender untuk memudahkan admin melihat total penjualan dari tanggal dan sampai tanggal, sehingga sistem akan menampilkan data barang terjual sesuai dengan kalender yang di atur oleh *admin*.



Gambar 4.35 Fitur Kalender

No	Item	Unit	Harga	Total
1
2
3
4
5
6
7
8
9
10

Gambar 4.36 Menu Barang Terjual

5. Halaman User

Menu *User* digunakan admin untuk menampilkan data *user* dan admin serta untuk tambah, edit dan hapus data *user* dan admin dari *database*.

No	ID User	Username	Name	Email	Password	Level	Aksi
1	1	admin	admin	[Edit] [Hapus]
2	2	user	User	[Edit] [Hapus]

Gambar 4.37 Halaman User

H. Pengujian Sistem

Pengujian dilakukan untuk mengetahui hasil proses sistem yang dibuat yaitu Sistem Pelaporan *Sparepart* Menggunakan *Barcode* Berbasis Web Studi Kasus Bengkel Maju Mandiri Servis Dan Autopart.

1. Pengujian *Black Box* Pada Sistem

Tabel 4.8 Pengujian *Black Box* Pada Sistem

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Saat Klik Menu Transaksi Penjualan	Klik Menu Transaksi Penjualan	Tampil Halaman Transaksi Penjualan	Sesuai Harapan	Valid
2	Saat Klik Menu Data	Klik Menu Data	Tampil Halaman Data Barang	Sesuai Harapan	Valid
3	Saat Klik Menu Barang Terjual	Klik Menu Barang Terjual	Tampil Halaman Barang Terjual	Sesuai Harapan	Valid
4	Saat Klik Menu Logout	Klik Menu Logout	Tampil Halaman Login	Sesuai Harapan	Valid

2. Pengujian *Black Box* Pada Transaksi Penjualan

Tabel 4.9 Pengujian *Black Box* Pada Transaksi Penjualan

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Saat Klik Transaksi Baru	Klik Transaksi Baru	Buat Kode Transaksi Baru Secara Otomatis	Sesuai Harapan	Valid
2	Saat Klik Tambah	Klik Tambah Setelah Scan Barcode atau Memasukkan Barcode Secara Manual	Form Secara Otomatis Diisi Sesuai Dengan Data Barang	Sesuai Harapan	Valid
3	Saat Klik Reset	Klik Reset	Hapus Barcode Pada Form Tambah	Sesuai Harapan	Valid
4	Saat Klik Lanjut Pada Form Tambah	Klik Lanjut	Form Tambah Masuk Kedalam Transaksi Sementara	Sesuai Harapan	Valid
5	Saat Klik Batal Pada Form Tambah	Klik Batal	Hapus Isi Form	Sesuai Harapan	Valid
6	Saat Klik Simpan Pada Transaksi Sementara	Klik Simpan	Transaksi Sementara Tersimpan dalam Barang Terjual sesuai dengan tanggal dan Stok Barang Berkurang	Sesuai Harapan	Valid

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
7	Saat Klik Batal Pada Transaksi Sementara	Klik Batal	Hapus Transaksi Sementara	Sesuai Harapan	Valid

3. Pengujian *Black Box* Pada Data Barang

Tabel 4.10 Pengujian *Black Box* Pada Data Barang

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Saat Klik Cari	Klik Cari Setelah Memasukkan ID Barang atau Nama Barang	Tampil Hasil Pencarian	Sesuai Harapan	Valid
2	Saat Klik Reset	Klik Reset Setelah Melakukan Pencarian	Hapus Pencarian	Sesuai Harapan	Valid
3	Saat Klik Logo Barcode	Klik Logo Barcode	Tampil Print Barcode	Sesuai Harapan	Valid
4	Saat Klik Edit	Klik Edit	Tampil Form Edit Data Barang	Sesuai Harapan	Valid
5	Saat Klik Hapus	Klik Hapus	Tampil Pop Up Pilihan "Ok" dan "Batal", Tampil Pop Up Data Barang Berhasil Dihapus	Sesuai Harapan	Valid
6	Saat Klik Next	Klik Next	Tampil Halaman Berikutnya	Sesuai Harapan	Valid
7	Saat Klik Prev	Klik Prev	Tampil Halaman Sebelumnya	Sesuai Harapan	Valid
8	Saat Klik Last	Klik Last	Tampil Halaman Terakhir	Sesuai Harapan	Valid
9	Saat Klik First	Klik First	Tampil Halaman Pertama	Sesuai Harapan	Valid
10	Saat Klik Tambah Data Barang	Klik Tambah Data Barang	Tampil Form Input Data Barang	Sesuai Harapan	Valid

4. Pengujian *Black Box* Pada Barang Terjual

Tabel 4.11 Pengujian *Black Box* Pada Barang Terjual

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Saat Klik Logo Kalender di Dari dan Sampai	Klik Logo Kalender di Dari dan Sampai	Tampil Hasil Data Barang Terjual Sesuai Tanggal Dari Dan Tanggal Sampai	Sesuai Harapan	Valid

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang dilakukan penulis, dapat memperoleh beberapa kesimpulan, yaitu :

1. Telah dibangun sebuah sistem untuk mempermudah kinerja dalam menghitung jumlah stok barang yang tersedia di Bengkel Maju Mandiri Servis Dan Autopart
2. Memberikan kemudahan kepada pengguna untuk melakukan pendataan secara terkomputerisasi dari pada melakukan pendataan secara manual yang memakan waktu.
3. Sistem ini sudah di uji menggunakan pengujian *black box testing* dan mendapatkan hasil yang baik.

B. Saran

1. Sistem dikembangkan lagi dari yang hanya untuk pelaporan hasil penjualan dan informasi stok barang jadi bisa untuk melakukan transaksi jual beli menggunakan barcode dan memiliki cetak struk pembelian.
2. Karena keterbatasan peneliti, tampilan sistem masih kurang bagus, dan peneliti berharap sistem ini dapat dikembangkan lagi menjadi lebih baik.

DAFTAR PUSTAKA

- Agung, A., Maulana, H. M., Andini, D. C., & Nadziroh, F. (2018). Sistem peminjaman ruangan online (spro) dengan metode uml. *Urnal Teknologi Dan Terapan Bisnis (JTTB)*, 1(1), 1–8.
- Content, M. C., & Informatika, M. (2015). (1) *Aplikasi Delivery Makanan Berbasis Web Di Area*. 1(1), 180–185.
- Enterprise, J. (2017). *Otodidak Mysql Untuk Pemula*. PT Elex Media Komputido.
- Halim, F., & Huwae, S. (2019). *Bengkel Motor Custom maka generasi Milenial berada di tingkat selanjutnya yaitu di tahap psychological needs . sesuai dengan keinginannya sehingga timbulnya kepuasan jiwa . perekonomian , terutama pada sektor riil , salah satu sektor riil yang sangat laya*. 1(2), 1735–1746.
- Heriyanto, Y. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car. *Jurnal Intra-Tech*, 2(2), 64–77.
- Hidayat, R., Marlina, S., & Utami, L. D. (2017). Perancangan Sistem Informasi Penjualan Barang Handmade Berbasis Website Dengan Metode Waterfall. *Simnasiptek*, 176–183.
- Hutahean, J. (2014). *Konsep Sistem Informasi*. Deepublish. https://books.google.co.id/books?hl=id&lr=&id=o8LjCAAAQBAJ&oi=fnd&pg=PR7&dq=pengertian+sistem&ots=t-sbukIViI&sig=AStZjIEe135D6BVfwwqpT-KaWVw&redir_esc=y#v=onepage&q&f=false
- Ir.Harianto Kristanto. (1994). *Konsep & Perancangan Database*. Andi. https://books.google.co.id/books?hl=id&lr=&id=tW0MAPGoaVAC&oi=fnd&pg=PA1&dq=definisi+database&ots=Qiq5OgKfSF&sig=eJFEyPd0C3rXzz37S0d0x3WrCs&redir_esc=y#v=onepage&q&f=false
- Morita, N., Shimizu, M., Arisawa, M., & Koshi, M. (1974). Studies on the Medicinal Resources. XXXVI. The Constituents of the Leaves of *Saxifraga stolonifera* Meerburg (Saxifragaceae). *Chemical and Pharmaceutical Bulletin*, 22(7), 1487–1489. <https://doi.org/10.1248/cpb.22.1487>

- Mulyono, A. (2019). Penerapan Barcode Scanner System Dalam Proses Pengujian Ban Di Laboratorium Uji Pt Xyz. *Teknologi : Jurnal Ilmiah Dan Teknologi*, 1(2), 80. <https://doi.org/10.32493/teknologi.v1i2.3048>
- Mustakim. (2018). *Pemograman Internet*. Cahaya Firdaus.
- Novianti, A., & Fauzijah, A. (2009). Sistem Informasi Sekolah Dasar Berbasis SMS. *Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009)*, 2009(Snati), A-40-A-45.
- Prasetiadi, A. E. (2011). *Web 3.0: Teknologi Web Masa Depan*. 1(3), 1–6.
- R.Mulyanto, A. (2008). *Rekayasa Perangkat Lunak*. Direktorat Pembinaan Sekolah Menengah Kejuruan. <https://dokumen.tech/reader/full/rekayasa-perangkat-lunak-jilid-1-56e4574286450>
- Sarmidi, S. I. R. (2018). Jurnal manajemen dan teknik informatika. *Rancang Bangun Sistem Informasi Pengolahan Bank Sampah Puspasari Kecamatan Purbaratu Kota Tasikmalaya*, 02(01), 181–190.
- Sidratul munti, N. yona. (2019). Perancangan Sistem Pakar Diagnosa Penyakit Lupus Eritmatusus Sistem(LES) Dengan Metode Forward Chaining Menggunakan Pemrograman PHP dan MySQL. *Jurnal Fasilkom*, 9(2), 407–428. <https://doi.org/10.37859/jf.v9i2.1404>
- Solichin, A. (2016). *Pemrograman Web dengan PHP dan MySQL*. Budi Luhur. https://books.google.co.id/books?hl=id&lr=&id=k8-GDAAAQBAJ&oi=fnd&pg=PR3&dq=dasar+web&ots=djbdC1yYbo&sig=xK6CaZOLN7Uj6tJFr9d9ZQcQ190&redir_esc=y#v=onepage&q=dasarweb&f=false