

BAB III

METODOLOGI PENELITIAN

3.1 Alur Penelitian

Pengembangan aplikasi Appbeng memiliki proses tahapan penelitian berdasarkan alur penelitian. Alur penelitian juga merupakan salah satu strategi penelitian agar tahapan proses penelitian lebih terstruktur dan dapat memaksimalkan waktu penelitian. Dapat dilihat pada Gambar 3.1 merupakan tahapan proses penelitian yang terdiri dari identifikasi masalah, studi literatur, pengembangan sistem, analisis dan evaluasi. Adapun penjabaran dari masing-masing tahapan adalah sebagai berikut :

1. Identifikasi masalah

Tahapan ini bertujuan untuk menentukan ruang lingkup permasalahan yang menjadi tujuan pembangunan aplikasi Appbeng. Permasalahan yang menjadi acuan adalah pengelolaan manajemen usaha masih dilakukan secara manual dan tidak adanya detail catatan tentang perputaran pengeluaran maupun pemasukan dan tidak adanya detail catatan *inventory* yang tersusun rapi.

2. Studi literatur

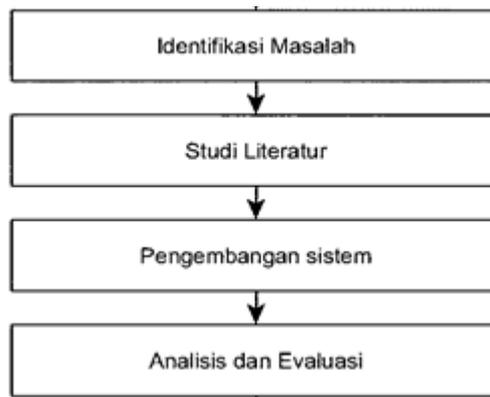
Tahapan ini bertujuan untuk mengumpulkan teori-teori yang berasal dari jurnal, buku, *website*, dan lainnya yang menjadi dasar pemikiran sebagai penunjang proses penelitian. Tahapan ini juga mencari dan mempelajari penelitian terkait sebagai referensi perbandingan (Tabel 2.1) dalam melakukan penelitian.

3. Pengembangan sistem

Tahapan selanjutnya merupakan tahapan pengembangan sistem sebagai implementasi solusi atas identifikasi masalah. Pengembangan sistem dimulai dari menganalisa kebutuhan sistem yang akan dibangun, perancangan tampilan sistem dan perancangan *database system*, kemudian implementasi dan pengujian.

4. Analisis dan evaluasi

Kemudian tahapan terakhir merupakan penjabaran hasil implementasi pengembangan sistem. Tahapan ini juga melakukan analisis pada hasil rekomendasi yang diberikan sistem dan juga respon sistem terhadap data yang masuk. Kemudian dilakukan evaluasi pada hasil rekomendasi dari penerapan metode *ABC cycle counting*. Evaluasi dilakukan untuk melihat pengaruh sistem yang telah dibangun terhadap bengkel UD Empat Senine.



Gambar 3.1 Alur penelitian

3.2 Pengembangan Sistem

Pengembangan sistem dilakukan menggunakan metode *agile development* dengan model *scrum*, sebelumnya dibutuhkan analisis kebutuhan untuk merancang pengerjaan aplikasi. Analisis membutuhkan keterlibatan *stakeholder* sebagai seseorang yang mengetahui lingkungan operasional baik yang dilakukan secara manual maupun menggunakan bantuan media digital.

3.2.1 Analisis kebutuhan

Analisa kebutuhan merupakan tahapan yang dilakukan untuk menentukan fungsi-fungsi fitur dalam pembangunan sistem. Kebutuhan didefinisikan dari batasan dan tujuan dari penelitian. Analisa juga dilakukan untuk menentukan kebutuhan fungsional dan nonfungsional pada aplikasi Appbeng.

3.2.1.1 Analisis Data

Penelitian ini membutuhkan beberapa data barang yang telah tercatat pada transaksi sebelumnya. Kemudian diperlukan analisis data yang saat ini digunakan pada UD Empat Senine. Berikut terdapat format pencatatan data keluar masuk barang yang

sebelumnya telah digunakan dalam bertransaksi, Dapat dilihat pada Tabel 3.1 merupakan contoh format catatan barang masuk dari *supplier*.

Tabel 3.1 Contoh format data barang masuk

No	Nama Barang	Harga Satuan	Jumlah	Total
1	Ban dalam 250x17 swl	Rp22.000	12	Rp264.000
	Ban dalam 275x17 swl	Rp21.500	12	Rp258.000
	Ban luar 250x17 swl	Rp115.000	5	Rp575.000
	Ban luar 275x17 swl	Rp120.000	5	Rp600.000
	Ban luar 80/90x14 swl	Rp130.000	4	Rp520.000
	Ban luar 90/90x14 swl	Rp140.000	5	Rp700.000
Total				Rp2.917.000
Jaya Abadi		08 Juni 2020		
2	Oli Yamalube matic 0,8L	Rp32.000	12	Rp384.000
	Oli Gear honda	Rp11.500	24	Rp276.000
	Oli Mesran 1L	Rp25.000	24	Rp600.000
	Oli MPX1 0,8L	Rp38.000	10	Rp380.000
	Oli MPX2 0,8L	Rp38.500	10	Rp385.000
	Oli Ultra tec 0,8L	Rp31.000	10	Rp310.000
Total				Rp2.335.000
UD Yudha		12 Juni 2020		

Data pada Tabel 3.1 merupakan salah satu *sample* format barang masuk UD Empat Senine yang terjadi pada tanggal 8 dan 12 juni 2020. Barang yang masuk merupakan dari 2 *supplier* yang berbeda, masing-masing *supplier* menyediakan jenis kategori barang ban dan oli. Kemudian terdapat contoh format data penjualan yang terjadi pada tanggal 15 Juni 2020, *sample* data tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh format riwayat penjualan

No	Tanggal	Nama barang	Jumlah	Harga Satuan	Total
1	15 Juni 2020	Ban dalam 80/90 x 14	2	Rp25.000	Rp50.000
		Busi Supra NGK	1	-	Rp15.000

No	Tanggal	Nama barang	Jumlah	Harga Satuan	Total
		Ban dalam 275x17 swl	1	-	Rp25.000
		Ban dalam 275x17 swl	1	-	Rp25.000
		Ban luar 90/90x14 swl	1	-	Rp150.000
		Baterai GS astra	1	-	Rp210.000
		Lahar 6301	3	Rp10.000	Rp30.000
		Gear 1 set supra 125	1	-	Rp140.000
		Ban dalam 275x17 swl	1	-	Rp25.000
		Bohlam Honda	1	-	Rp25.000
Total					Rp695.000

Data transaksi penjualan yang tercatat merupakan catatan data barang yang terjual pada setiap transaksi per hari. Pencatatan tidak memiliki nomor *invoice* dan harga barang tidak termasuk biaya pemasangan barang oleh mekanik. Beberapa kriteria yang akan digunakan dalam tahap pengumpulan data adalah sebagai berikut :

- a) Pencatatan barang masuk merupakan catatan dari nota belanja dari *supplier* barang. Informasi dari tiap *supplier* akan dibedakan untuk disusun pada sistem agar terlihat lebih rapi.
- b) Kategori data barang merupakan pencatatan dari barang yang sejenis, barang tersebut terkumpul dari beberapa jenis SKU dengan berbagai spesifikasi.
- c) Pengecekan stok barang akan dilakukan pada saat implementasi sistem, hal ini dikarenakan stok barang yang bersifat dinamis. Pengecekan yang dilakukan berdasarkan *sample random* memiliki resiko terhadap perbedaan stok dan sistem dikhawatirkan kurang optimal dalam memberikan rekomendasi.
- d) Pengumpulan data riwayat transaksi menjadi acuan dalam menentukan frekuensi permintaan barang. Data frekuensi permintaan barang akan dibutuhkan untuk pengolahan sistem rekomendasi.

3.2.1.2 Kebutuhan Pengguna

Pada sub bagian 3.2.1.2, akan dijelaskan analisis kebutuhan sistem berdasarkan kebutuhan pengguna. Pengembangan aplikasi Appbeng memiliki 2 *role* pengguna,

yaitu Admin dan Operator. Berdasarkan analisis ini, akan ditentukan kebutuhan fungsional dan nonfungsional sistem. Pada Tabel 3.3 akan dijelaskan kebutuhan sistem berdasarkan pengguna.

Tabel 3.3 Analisis kebutuhan pengguna

No	Pengguna	Deskripsi
1	Admin	Admin mengelola pelayanan dan pengelolaan <i>inventory</i> usaha. Admin juga dapat melihat rekomendasi produk yang diberikan sistem, memanipulasi data barang, data pelanggan, data supplier, riwayat transaksi, dan melihat keuntungan penjualan.
2	Operator	Operator bertugas sebagai mencatat hasil penjualan dan dapat melihat riwayat transaksi penjualan.

3.2.1.3 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fungsi-fungsi yang dapat dilakukan oleh aplikasi Appbeng. Penentuan kebutuhan fungsional ditentukan berdasarkan analisis penggunaan dari aplikasi. Untuk menentukan kebutuhan aplikasi Appbeng, diperlukan pengumpulan informasi seperti wawancara yang terlampir pada Lampiran A. Berdasarkan dari hasil wawancara tersebut, penulis membagi 8 fitur yang dapat dilakukan Admin dan 2 fitur yang dapat dilakukan Operator. Pada Tabel 3.4 akan diuraikan kebutuhan fungsional sistem pada aplikasi Appbeng.

Tabel 3.4 Kebutuhan fungsional sistem

No.	Kebutuhan fungsional	Deskripsi
Admin		
1.	Fitur <i>log in</i>	Admin memerlukan <i>username</i> dan <i>password</i> untuk masuk ke halaman dashboard sistem untuk dapat menggunakan fitur pengelolaan aplikasi Appbeng.
2.	Fitur <i>log out</i>	Admin dapat keluar dari sistem untuk memutuskan hak akses yang digunakan.

No.	Kebutuhan fungsional	Deskripsi
3.	Rekomendasi produk	Fitur ini memberikan rekomendasi terhadap Admin untuk melakukan tindakan pada barang yang dianggap menjadi prioritas. Pemberian rekomendasi dilakukan oleh metode <i>ABC Cycle Counting</i> .
4.	Klasifikasi barang	Melakukan klasifikasi produk berdasarkan data transaksi penjualan, barang yang dianggap memiliki pengaruh besar akan dibedakan berdasarkan <i>class</i> .
5.	Pengecekan <i>physical inventory</i>	Memberikan jumlah pengecekan <i>inventory</i> secara berkala sesuai dengan tingkat kepentingan suatu produk.
6.	Riwayat transaksi	Menampilkan catatan transaksi barang keluar dan barang masuk.
7.	Data pelanggan	Mengelola data pelanggan, dimulai dari menambah, mengedit, dan menghapus data pelanggan.
8.	Data <i>service</i>	Mengelola data pelanggan yang melakukan servis kendaraan.
9.	Data <i>inventory</i>	Mengelola data produk usaha, pengelolaan meliputi menambah, mengedit, dan menghapus produk pada <i>inventory</i> .
10.	Laba keuntungan	Menampilkan laba keuntungan pada <i>range</i> tanggal yang ditentukan oleh <i>user</i> .
11.	Data <i>supplier</i>	Informasi data penyuplai barang tercatat pada menu data <i>supplier</i> . Pencatatan dilakukan untuk pengarsipan data <i>supplier</i> agar lebih rapi.
12	<i>Purchase order</i> .	Melakukan pemesanan pendataan barang yang dibeli dari <i>supplier</i> .
Operator		

No.	Kebutuhan fungsional	Deskripsi
13.	Transaksi pembayaran	Mencatat transaksi pembayaran dan penjualan produk.
14.	Riwayat transaksi	Menampilkan riwayat transaksi penjualan.

3.2.1.4 Kebutuhan Nonfungsional

Kebutuhan nonfungsional menjadi batasan layanan atau fungsi pada sistem. Secara umum kebutuhan ini terbagi menjadi empat macam, yaitu *usability*, *portability*, *reliability*, dan *supportability*. Pada aplikasi Appbeng, terdapat beberapa kebutuhan nonfungsional yang dapat dilihat pada Tabel 3.5.

Tabel 3.5 Kebutuhan nonfungsional sistem

No	Kebutuhan nonfungsional
1.	Sistem dapat berjalan di <i>web browser google chrome</i> .
2.	Sistem memiliki UI/UX yang <i>simple</i> dan mudah dipahami.
3.	Sistem memiliki tampilan yang <i>responsive</i> , terutama di tablet dan pc.
4.	Waktu pengaksesan halaman <i>website</i> tidak melebihi <i>5 second</i> .

3.2.2 Metodologi Pengembangan

Dalam melakukan pengembangan dengan menggunakan model *scrum*, diperlukan pembentukan tim dalam pelaksanaannya. Tim *scrum* terdiri dari *product owner*, *development team*, dan *scrum master*. Adapun penjelasan tugas dari tim *scrum* untuk pengembangan aplikasi Appbeng adalah sebagai berikut :

1. *Product Owner*

Bertanggung jawab untuk memaksimalkan *value* dari perangkat lunak yang akan dibangun dan melakukan *review* terhadap hasil *sprint* atau menyampaikan jika terdapat tambahan kebutuhan fitur.

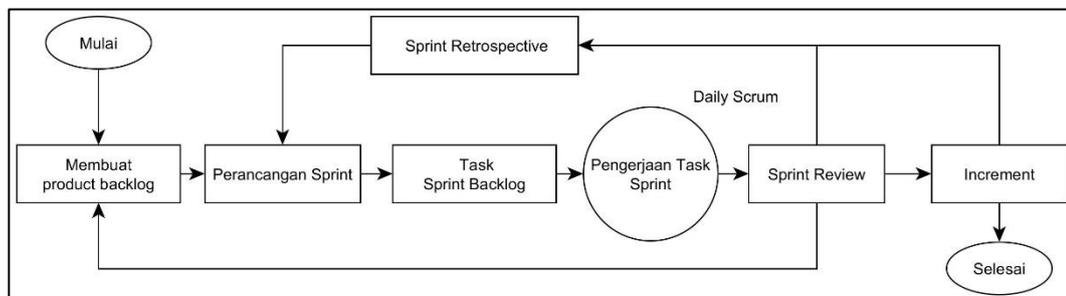
2. *Scrum Master*

Bertanggung jawab untuk memastikan penerapan *scrum* dijalankan dengan baik, dimulai dari aktivitas *sprint* maupun aktivitas *scrum* lainnya.

3. *Development Team*

Bertanggung jawab dalam pengembangan fitur-fitur yang akan dibangun pada aplikasi.

Berdasarkan penjelasan tugas tim *scrum*, pengembangan aplikasi Appbeng terdiri dari Helmi Virgi Yanto, S.E. sebagai *product owner* sekaligus pemilik usaha UD Empat senine dan penulis sendiri sebagai *scrum master* sekaligus *development team*. *Scrum* memiliki alur kegiatan yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram alur pengembangan sistem

Berdasarkan Gambar 3.2 menunjukkan bahwa kegiatan pengembangan dengan menggunakan model *scrum* meliputi pembuatan *product backlog*, perancangan *sprint*, membuat *task sprint*, pengerjaan *task sprint*, dan *sprint review*. Kemudian jika sistem terdapat tambahan dari *product backlog*, maka proses tersebut masuk ke proses *increment*, yaitu sesuatu tambahan dari *product backlog* yang terjadi pada saat *sprint* berjalan. Kemudian masuk ke tahapan *sprint retrospective* untuk melakukan evaluasi agar *sprint* berikutnya menjadi lebih baik lagi. Siklus ini akan terus berulang hingga semua *sprint* berhasil diselesaikan. Rancangan product backlog pada pengembangan aplikasi Appbeng dapat dilihat pada Tabel 3.6.

Tabel 3.6 *product backlog*

No	Task Backlog	Estimasi (Jam)	Tingkat Kesulitan	Prioritas	Scrum Sequence
1	Analisis kebutuhan	25	Rendah	Tinggi	1
2	Mengumpulkan <i>resource</i>	10	Rendah	Sedang	1
3	Merancang UI/UX	25	Tinggi	Sedang	1
4	Implementasi <i>mockup</i>	25	Tinggi	Tinggi	2
5	Merancang <i>database</i>	25	Tinggi	Tinggi	2
6	Pembuatan fungsi <i>login</i>	10	Rendah	Sedang	2
7	Pembuatan menu <i>dashboard</i>	10	Sedang	Sedang	3

No	Task Backlog	Estimasi (Jam)	Tingkat Kesulitan	Prioritas	Scrum Sequence
8	Pembuatan menu <i>Supplier</i>	5	Rendah	Sedang	3
9	Pembuatan menu Data pelanggan	15	Sedang	Sedang	3
10	Pembuatan menu <i>service</i>	25	Tinggi	Tinggi	3
11	Pembuatan menu data produk	20	Tinggi	Tinggi	3
12	Pembuatan menu kategori	15	Sedang	Rendah	3
13	Pembuatan menu <i>purchase order</i>	30	Tinggi	Tinggi	4
14	Pembuatan menu riwayat	5	Sedang	Tinggi	4
15	Pembuatan menu <i>filter</i> pendapatan	15	Tinggi	Tinggi	4
16	Pembuatan menu klasifikasi	30	Tinggi	Tinggi	4
17	Pembuatan menu pengecekan	25	Tinggi	Tinggi	4
18	Implementasi rumus ABC analisis	15	Tinggi	Tinggi	4
19	Pembuatan fungsi notifikasi	10	Sedang	Sedang	5
20	Pembuatan fungsi <i>logout</i>	10	Rendah	Sedang	5
21	Pembuatan menu beranda	30	Tinggi	Tinggi	5
22	Pengujian aplikasi	10	Sedang	Tinggi	5
Total Jam kerja		390 Jam			

Implementasi pengerjaan *scrum* memiliki 5 hari jam kerja setiap minggunya dengan jam kerja 6 jam per hari. Pembentukan *sprint* akan dilakukan berdasarkan *scrum sequence* secara berurut. Selanjutnya dibangun *gant chart* untuk membentuk rancangan *timeline* dari pengembangan aplikasi Appbeng. Dapat dilihat pada Tabel 3.7 merupakan *gant chart* dari *sprint* dengan waktu pengerjaan dimulai dari bulan

Januari minggu pertama hingga ke bulan April minggu pertama. Berdasarkan *scrum sequence product backlog*, pengembangan aplikasi Appbeng terbagi menjadi 5 *sprint*.

Tabel 3.7 Gant Chart

Nama kegiatan	Januari				Februari				Maret				April
	1	2	3	4	1	2	3	4	1	2	3	4	1
<i>Sprint 1</i>													
<i>Sprint 2</i>													
<i>Sprint 3</i>													
<i>Sprint 4</i>													
<i>Sprint 5</i>													

3.3 Rancangan Implementasi Metode ABC Cycle Counting

Dalam merancang implementasi metode ABC *cycle counting*, diperlukan beberapa tahapan yaitu analisis frekuensi, klasifikasi *class*, dan analisis pemilihan sistem rekomendasi. Adapun masing-masing dari penjelasan tahapan adalah sebagai berikut :

1. Analisis Frekuensi

Analisis frekuensi merupakan analisis yang melakukan perhitungan data dari hasil penjualan. Adapun perhitungan yang dilakukan adalah dengan cara menghitung jumlah kemunculan dari masing-masing produk. Perhitungan dilakukan tanpa menghiraukan jumlah dan harga produk. Nilai dari jumlah masing-masing produk disebut nilai frekuensi permintaan barang. Hal ini memudahkan pemilik usaha ketika ingin mengetahui tingkat kelarisan barang dan tidak perlu membeli persediaan barang dengan jumlah stok setiap produk sama rata. Untuk melakukan analisis frekuensi, dapat menerapkan dengan menggunakan rumus persamaan 2.2

2. Klasifikasi Class

Sebelum melakukan klasifikasi *class*, dipastikan nilai total dari frekuensi permintaan barang telah didapat. Pengklasifikasian bertujuan untuk memisahkan produk dalam satu kelompok berdasarkan tingkat perputaran barang. Secara umum pengklasifikasian dibagi menjadi 3, yaitu A, B, dan C. *Class A* merupakan barang

yang memiliki tingkat perputaran cepat, *class* B merupakan barang yang memiliki tingkat perputaran sedang, dan *class* C merupakan barang yang memiliki tingkat perputaran rendah. Adapun cara yang digunakan untuk melakukan klasifikasi *class* adalah menghitung persentase untuk setiap *item* barang menggunakan rumus persamaan 2.1.

Kemudian dilakukan pengurutan produk yang telah memiliki nilai persentase produk berdasarkan nilai persentase dari terbesar hingga terkecil. Selanjutnya daftar produk yang telah diurutkan dilakukan perhitungan kumulatif persentase produk dengan menggunakan rumus persamaan 2.3. Nilai persentase produk akan menjadi acuan dalam melakukan klasifikasi ke dalam *class* A, B, C secara berturut-turut dengan masing-masing bobot nilai persentase produk 60-80%, 15-20%, dan 5-10%.

3. Analisis sistem rekomendasi

Dalam melakukan pemberian rekomendasi produk ke pengguna, sistem akan melakukan analisis dari hasil pengklasifikasian *class*. Analisis ini berdasarkan observasi dari kondisi yang terjadi pada UD Empat Senine. Penulis menerapkan pemberian batasan stok pada masing-masing *class*. Misalkan produk pada *class* A, sistem akan memberikan rekomendasi ketika produk tersisa 15 unit, kemudian pada *class* B sistem akan memberikan rekomendasi ketika produk tersisa 10 unit, dan pada *class* C sistem akan memberikan rekomendasi ketika produk tersisa 5 unit.

3.4 Desain Sistem

Desain sistem merupakan tahapan dalam merancang implementasi sistem yang akan dibangun. Rancangan didasari oleh kebutuhan yang telah dianalisis, analisis kebutuhan tersebut dilakukan dengan cara pengumpulan data, observasi, dan wawancara dengan *product owner* yang sekaligus merupakan pemilik usaha UD Empat Senine. Desain sistem yang dibangun akan menjadi acuan dalam proses *development* aplikasi Appbeng.

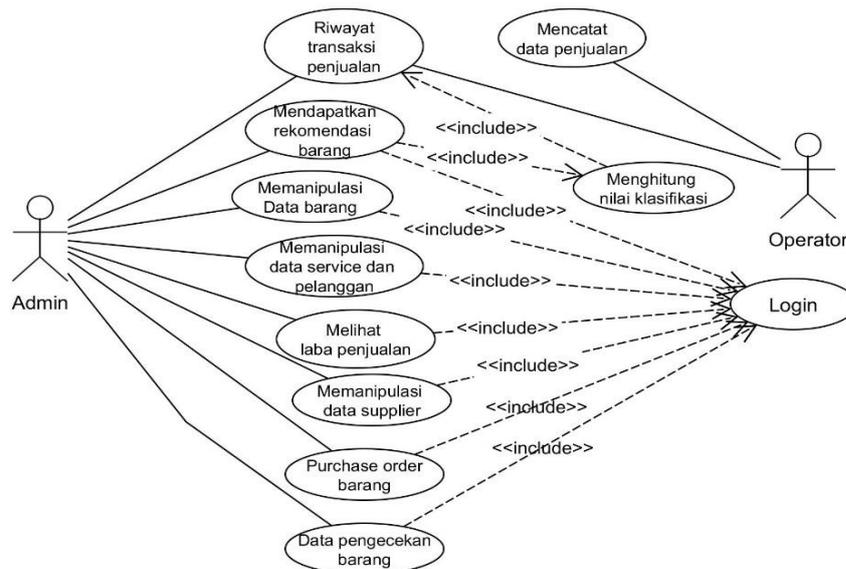
3.4.1 Perancangan Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *object-oriented* [22]. UML juga menjadi bahasa yang digunakan untuk mempermudah

seseorang yang tidak memiliki *basic* IT dalam memahami alur aplikasi yang akan dibangun.. Jenis dari UML yang akan dibangun yaitu *Use Case*, *Activity Diagram* dan *Class Diagram*.

1. *Use Case Diagram*

Aplikasi Appbeng memiliki rancangan dua pengguna, yaitu Admin dan Operator. Admin adalah seseorang yang memegang hak kendali penuh pada sistem terhadap pengelolaan bengkel, dimulai dari mengelola data produk hingga data *service* kendaraan pelanggan dan *supplier*. Selain itu seorang Admin memiliki akun untuk masuk ke system pengelolaan, dan Operator adalah yang bertanggung jawab dalam melakukan pencatatan transaksi keluar dan dapat melihat riwayat transaksi. Selain itu, seorang Operator tidak memerlukan *login* untuk masuk ke sistem. Untuk melihat gambar dari rancangan pengguna tersebut, dapat dilihat pada Gambar 3.3 yang merupakan *use case diagram* aplikasi Appbeng.



Gambar 3.3 *Use Case Diagram*

Pada rancangan *use case diagram*, pengguna Admin adalah orang yang bertanggung jawab dalam mengelola sistem aplikasi yang diharuskan *login* terlebih dahulu untuk dapat mengakses fiturnya. Adapun fitur tersebut yaitu data pengecekan barang, *purchase order* barang, memanipulasi data *supplier*, memanipulasi data barang, melihat laba penjualan, mendapatkan rekomendasi barang, dan riwayat penjualan. Sedangkan pengguna Operator adalah pengguna

yang bertanggung jawab melayani transaksi barang, Operator memiliki fitur dapat mencatat data penjualan dan melihat riwayat.

2. *Activity Diagram*

Activity diagram dibangun untuk menggambarkan urutan kegiatan yang terjadi pada saat menggunakan aplikasi. Diagram ini akan menjelaskan penggunaan aplikasi secara keseluruhan yang terbagi menjadi 2 pengguna, yaitu Admin dan Operator. Seperti terlihat pada Gambar 3.4 yaitu alur kegiatan berupa *activity diagram* dari Admin. Pada gambar tersebut dijelaskan bahwa Admin memiliki 8 fitur utama yang dapat diakses, yaitu :

A. *Dashboard*

Menu *dashboard* merupakan halaman utama pengguna Admin, berisikan ringkasan pendapatan hari ini, total produk, total supplier, dan total pelanggan yang terdaftar pada aplikasi Appbeng. Selain itu menu ini juga menampilkan daftar produk yang harus dilakukan *restock* berdasarkan hasil klasifikasi yang telah dilakukan.

B. *Supplier*

Menu *supplier* merupakan data *supplier* produk, dapat dilakukan penambahan, mengubah, dan menghapus data *supplier*.

C. Data Pelanggan

Menu data pelanggan dapat mengelola pelanggan yang melakukan *service* kendaraan di bengkel UD Empat Senine. Selain itu, menu ini dapat melakukan pengelolaan *service* ke masing-masing pelanggan.

D. Data Produk

Menu data produk merupakan pendataan produk yang dijual, menu dapat melakukan penambahan, penghapusan, mengubah, dan melihat detail produk.

E. Kategori

Menu kategori merupakan membantu pengelompokan data produk, kategori juga dapat membantu barang lebih terorganisir dan dapat memudahkan melakukan penyaringan berdasarkan kelompok barangnya.

F. *Purchase Order*

Menu *purchase order* merupakan menu pencatatan pesanan terhadap *supplier*, produk yang dipesan akan didata terlebih dahulu sesuai dengan jumlah, jenis

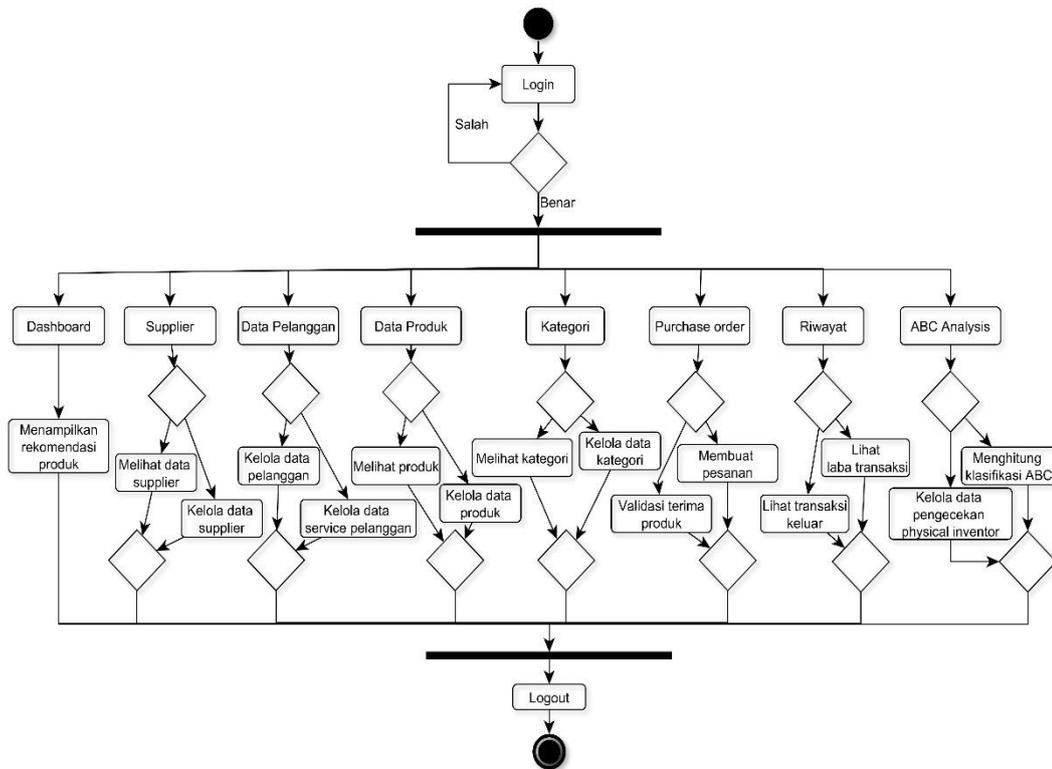
produk, dan harga yang menyesuaikan dengan nota yang ada pada *supplier*, kemudian setelah produk telah sampai dan diterima oleh pihak bengkel maka dilakukan verifikasi telah penerimaan barang dan menu ini mencatat bahwa pembelian produk dari *supplier* berhasil.

G. Riwayat

Menu riwayat merupakan halaman riwayat transaksi keluar, menu ini juga dapat melakukan perhitungan keuntungan berdasarkan tanggal yang dipilih oleh *user*.

H. ABC Analysis

Menu ABC analysis terdiri dari 2 sub menu, yaitu klasifikasi dan pengecekan. Klasifikasi merupakan menu yang melakukan klasifikasi ABC berdasarkan transaksi keluar dengan rentang waktu yang ditentukan oleh *user*, kemudian hasil klasifikasi akan melakukan *update* data ke menu pengecekan. Data yang diperbaharui oleh menu tersebut merupakan jumlah pengecekan *physical inventory* berdasarkan kelasnya.



Gambar 3.4 Activity Diagram Admin

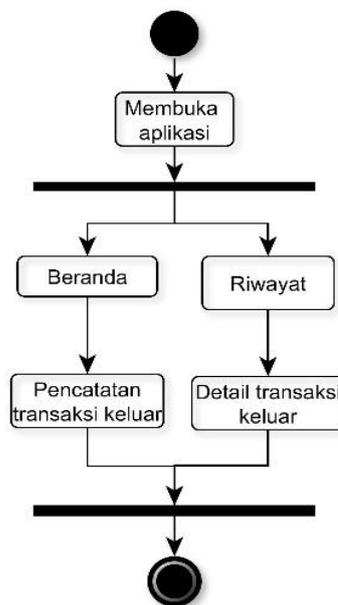
Sedangkan pada Gambar 3.5 merupakan *activity diagram* dari kegiatan operator. Adapun menu yang dapat dijalankan operator adalah sebagai berikut :

A. Beranda

Menu beranda merupakan halaman pencatatan penjualan oleh operator, dilakukan dengan memasukkan kode produk dan memasukkan jumlah dan harga yang dibayar, kemudian lakukan submit untuk melakukan pembelian.

B. Riwayat

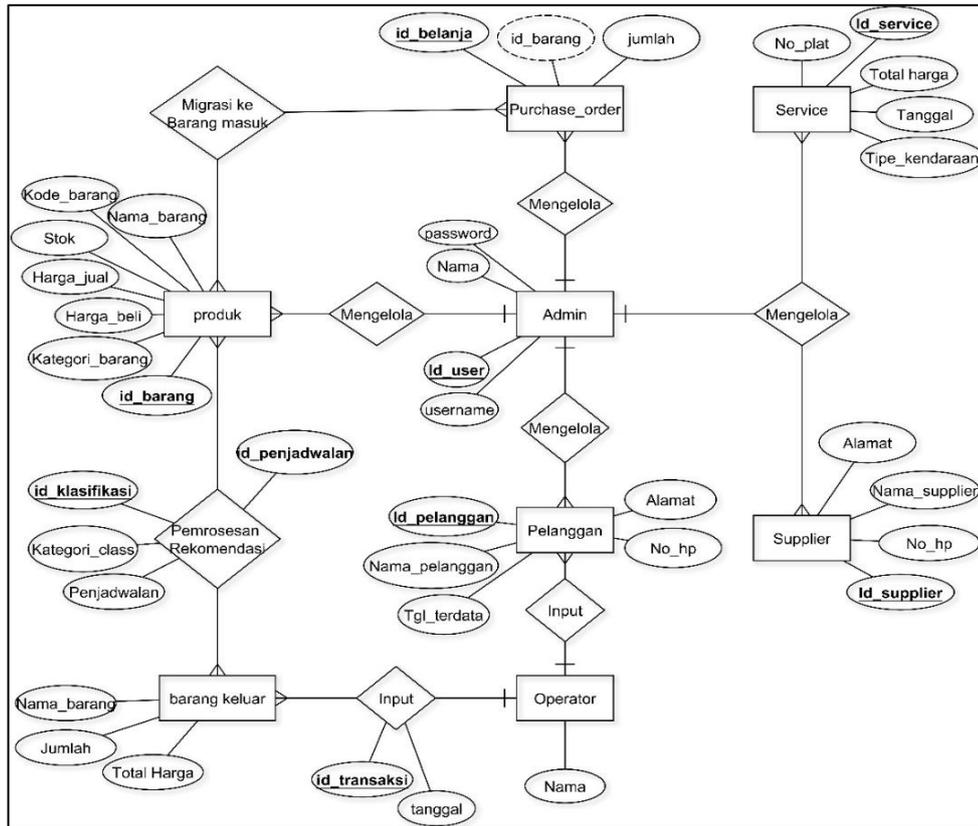
Menu ini untuk melihat riwayat transaksi keluar yang telah terjadi, dan tidak dapat melihat laba keuntungan penjualan melalui filter tanggal.



Gambar 3.5 *Activity Diagram* Operator

3.4.2 Perancangan Entity Relationship Diagram (ERD)

ERD menggambarkan relasi antar entitas serta atribut *database* sistem aplikasi Appbeng sebagai proses pengolahan penyimpanan data. Entitas pada aplikasi Appbeng memiliki 8 entitas, yaitu admin, pelanggan, operator, service, barang keluar, produk, dan daftar belanja. Adapun entitas admin merupakan pengelola aplikasi Appbeng yang melakukan pengelolaan terhadap barang masuk, produk, supplier, data servis, dan data pelanggan, sedangkan operator merupakan pengelola terhadap barang keluar, dan input data pelanggan. Dapat dilihat pada Gambar 3.6 merupakan ERD dari sistem aplikasi Appbeng.



Gambar 3.6 Diagram ERD sistem

3.5 Desain Pengujian Sistem

3.5.1 Desain Pengujian *Blackbox*

Pengujian aplikasi Appbeng akan dilakukan menggunakan metode *blackbox* dengan teknik *equivalence partitions*. Berdasarkan kebutuhan pengguna, pengujian sistem dibagi menjadi bagian pengujian sebagai operator dan pengujian sebagai admin. Uraian skenario pengujian sistem sebagai operator dapat dilihat pada Tabel 3.8.

Tabel 3.8 Skenario pengujian sistem sebagai operator

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
1	Beranda Transaksi	Pengujian melakukan fitur pembelian transaksi produk dan mengklik tombol submit	Transaksi berhasil dilakukan dan sistem menyimpan data transaksi

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
2	Riwayat Transaksi	Melihat riwayat transaksi yang telah terjadi di tanggal tertentu	Sistem menampilkan transaksi pada tanggal yang telah dipilih
		Melihat detail produk riwayat transaksi	Menampilkan detail produk transaksi yang dipilih

Selanjutnya terdapat pengujian sistem sebagai admin, adapun skenario pengujian yang telah dirancang dapat dilihat pada Tabel 3.9.

Tabel 3.9 Skenario pengujian sistem sebagai admin

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
1.	<i>Login</i>	Mengisi <i>e-mail</i> dan <i>password</i> yang terdaftar dengan benar	<i>User</i> berhasil masuk ke sistem sebagai admin
		Mengisi <i>e-mail</i> dan <i>password</i> yang salah	Sistem memberikan pesan “ <i>These credentials do not match our records.</i> ” dan tidak berhasil masuk ke sistem
2.	Notifikasi	Menampilkan produk rekomendasi <i>restock</i>	Menampilkan produk yang masuk ke daftar pembelian
3.	<i>Dashboard</i>	Melihat ringkasan jumlah data dan daftar pembelian rekomendasi produk	Sistem menampilkan produk yang direkomendasikan untuk penambahan stok
4.	<i>Supplier</i>	Melakukan penambahan data <i>supplier</i>	Data <i>supplier</i> berhasil ditambah dan tersimpan pada <i>database</i>
		Menghapus data <i>supplier</i>	Data <i>supplier</i> berhasil dihapus
		Mengubah data <i>supplier</i>	Data <i>supplier</i> yang dipilih berhasil diubah
5.	Data pelanggan	Menambah data pelanggan	Data berhasil ditambah dan tersimpan di <i>database</i>

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
		Mengubah data pelanggan	Data berhasil diubah dan tersimpan di <i>database</i>
		Menghapus data pelanggan	Data yang dipilih berhasil dihapus
6.	Data <i>service</i> kendaraan	Menambah data <i>service</i> pelanggan	Data <i>service</i> berhasil ditambah ke pelanggan yang telah dipilih
		Menghapus data <i>service</i> pelanggan	Data produk berhasil ditambah dan tersimpan <i>pada</i> database
7.	Produk	Menambah data produk	Data produk berhasil ditambah dan tersimpan <i>pada</i> database
		Mengubah data produk	Produk yang dipilih telah berhasil diubah
		Detail data produk	Sistem menampilkan detail data produk
		Menghapus data produk	Produk yang dipilih berhasil dihapus
8.	Kategori	Menambah kategori produk	Kategori baru berhasil disimpan dan tersimpan <i>pada</i> <i>database</i>
		Mengubah kategori produk	Data kategori berhasil <i>diupdate</i>
		Menghapus kategori produk	Kategori yang dipilih berhasil terhapus
9.	<i>Purchase order</i>	Membuat pesanan <i>purchase order</i> dari <i>supplier</i>	Data <i>purchase order</i> telah berhasil dibuat dan PO berstatus <i>pending</i>
		Melihat detail pesanan <i>purchase order</i>	Sistem menampilkan produk dengan jumlah yang dipesan
		Mengubah atau menghapus produk pesanan PO yang berstatus <i>pending</i>	Data pesanan <i>purchase order</i> telah berubah dan data berhasil <i>diupdate</i>

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
		Memvalidasi penerimaan barang pesanan PO	Status barang pesanan berubah menjadi complete dan stok produk bertambah sesuai jumlah produk pesanan
		Menghapus pesanan <i>Purchase order</i>	Pesanan PO berhasil dihapus
10.	Riwayat Transaksi	Melihat pendapatan transaksi yang telah terjadi di tanggal tertentu	Sistem menampilkan pendapatan dengan range tanggal yang telah dipilih
		Melihat detail produk riwayat transaksi	Menampilkan detail data produk pada transaksi yang telah terjadi
11.	<i>Logout</i>	Keluar dari halaman sistem admin	<i>User</i> keluar dari halaman sistem admin
Sistem Rekomendasi			
12.	Klasifikasi	Menghitung nilai klasifikasi class produk	Sistem berhasil menambah data klasifikasi baru dan tersimpan pada database, kemudian <i>class</i> pada produk akan diupdate sesuai dengan nilai persentase yang telah dihitung
		Menghapus klasifikasi yang pernah dihitung	Data klasifikasi dan hasil detail klasifikasi telah berhasil dihapus
13	Pengecekan <i>physical inventory</i>	Melakukan pengecekan <i>physical inventory</i> saat status insialisasi awal	Jumlah status pengecekan akan berkurang sebanyak 1x dari jumlah pengecekan, jumlah pengecekan akan menyesuaikan dengan class produk
		Pengecekan <i>physical inventory</i> dilakukan sudah selesai dilakukan	Status pengecekan akan berubah Kembali sebanyak jumlah pengecekan yang telah

No	Fungsi	Deskripsi pengujian	Hasil yang diharapkan
		sebanyak jumlah yang diberikan	ditentukan klasifikasi <i>class</i> produk

3.5.2 Desain pengujian waktu eksekusi

Pengujian waktu eksekusi dibutuhkan sebagai pengujian kebutuhan nonfungsional untuk mengetahui kecepatan *website* dalam menampilkan seluruh elemen. Pengujian menggunakan *webttool audit website* yang telah disediakan oleh Google yaitu *Google Pagespeed Insight (PSI)*. Selain itu, teknologi yang digunakan menggunakan *lighthouse* untuk mengukur performa situs web. Teknologi PSI menggunakan *Chrome User Experience (CrUX)*, yaitu mempelajari pengalaman dari pengguna *chrome*. *CrUX* melaporkan 2 metrik yang disebut dengan *First Contentful Paint (FCP)* dan *DOMContentLoaded (DCL)*. Adapun penjelasan *FCP* dan *DCL* adalah sebagai berikut :

1. *First Contentful Paint (FCP)* adalah metrik yang melaporkan waktu saat browser pertama kali merender teks, gambar, kanvas non-putih, atau SVG.
2. *DOMContentLoaded (DCL)* adalah metrik yang melaporkan waktu ketika dokumen HTML awal telah sepenuhnya dimuat dan diurai, tanpa menunggu *stylesheet*, gambar, dan *subframe* selesai memuat.

Dari penjelasan tersebut, pemilihan *Google Pagespeed Insight* sebagai *webttool* pengujian performa situs *web* sudah tepat, hal ini dikarenakan *Pagespeed Insight* dikeluarkan oleh salah satu mesin pencarian yang digunakan seluruh dunia. Adapun Langkah-langkah pengujian dilakukan sebagai berikut :

1. Sebelum membuka halaman *webttool*, aplikasi *website* terlebih dahulu di *upload* ke penyedia *hosting*, kemudian masukkan alamat berikut <https://developers.google.com/speed/pagespeed/insights/> ke *browser*.
2. Salin *url* halaman aplikasi, kemudian tempelkan di kolom yang disediakan.
3. Selanjutnya klik tombol analisis, lakukan proses ini hingga proses semua halaman dianalisis.
4. Catat nilai skor dan *time interactive* masing-masing halaman yang diberikan *webttool*. Kemudian cari rata-rata dari nilai skor dan *time interactive*.