BAB II

LANDASAN TEORI

2.1 Pengertian Bengkel

Sistem manajemen bengkel yang baik diawali dengan pencatatan sederhana yang teratur, rapi, sistematis, serta aktual. Dari pencatatan sederhana tersebut kemudian dilakukan sejumlah analisis yang hasilnya akan mempengaruhi ritme atau proses kerja dari unit yang lain.

Untuk meningkatkan efektivitas dan efisiensi operasional dalam bengkel mulai dari sejak konsumen datang membawa kendaraannya, sampai pada selesainya kendaraan dilakukan serah terima kembali ketangan konsumen memuat rangkaian kejadian yang terlupakan oleh konsumen, yang harus dijaga kualitasnya, sehingga kesan puas justru tertanam dalam benak pelanggan anda (Utomo.2010).

Berikut ini adalah contoh alur proses bisnis standar yang ada dibengkel :

1. Proses pelayanan dimulai ketika pelanggan datang ke bengkel dengan membawa kendaraan miliknya yang ingin diperbaiki.
2. *Service Advisor* (SA) akan menyambut pelanggan lalu menanyakan dan memeriksa keluhan-keluhan kerusakan dari si pelanggan. SA kemudian memperkirakan apakah itu perlu diperbaiki atau cukup dilakukan pengecekan saja.
3. Jika terdapat kerusakan bodi, SA menanyakan apakah kendaraan pelanggan tersebut diasuransikan atau tidak.
4. Apabila pelanggan sudah setuju dengan estimasi harga yang ditawarkan maka SA akan membuat selembar acuan kerja mekanik bengkel rangkap tiga (satu

untuk pelanggan, satu untuk SA, satu untuk montir) sehingga kendaraan tersebut dapat segera dikerjakan.

5. Setelah acuan kerja tersebut keluar maka tanggung jawab perbaikan kendaraan diserahkan kepada mekanik dan SA.
6. Jika hasil pekerjaan telah selesai maka akan dibuatkan laporan dari perbaikanyang telah dilakukan dan data waktu perbaikan.
7. Apabila proses pekerjaan perbaikan telah selesai maka kendaraaan akan diserahkan kepada SA untuk diinformasikan kepada pelanggan.

Sebuah bengkel melakukan pencatatan berapa jumlah pelanggan perhari dan apa saja transaksi yanng dilakukan. Catatan tersebut dapat menjelaskan seberapa besar potensi pasar yang berhasil dikerjakan per harinya, onderdil apa saja yang serig mereka beli, jasa *service* apa saja yang sering mereka minta, serta mengapa hanya jasa *service* tersebut yang meraka minta, hal ini akan berkaitan dengan onderdil apa saja yang sebaiknya dipasok dalam jumlah banyak. Selain itu, apa saja yang boleh dipasok dalam jumlah sedikit dan berapa bulan sekali harus melakukan pengadaan baru sebelum pasokan yang lama benar-benar habis. Hal tersebut dapat dilihat dari pencatatan stok yang baik.

2.2 Penjualan

2.2.1 Pengertian Penjualan

Definisi penjualan menurut Mulyadi (2008:202), “Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli”. Penjualan merupakan

sumber hidup suatu perusahaan, karena dari penjualan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan untuk mengetahui daya tarik konsumen sehingga dapat mengetahui hasil produk yang dihasilkan.

2.2.2 Jenis Penjualan

Menurut Martin, dkk (2006), penjualan dapat dibedakan dan diidentifikasikan dari perusahaannya, antara lain:

1. Penjualan Langsung, yaitu penjualan dengan mengambil barang dari supplier dan langsung dikirim ke pelanggan.
2. Penjualan Stok Gudang, yaitu penjualan barang dari stok yang telah tersedia di gudang.
3. Penjualan Kombinasi, yaitu penjualan dengan mengambil barang yang sebagian dari supplier dan sebagian dari stok yang tersedia di gudang.

2.2.3 Sistem Penjualan

Sistem Penjualan adalah sekelompok unsur atau bagian yang saling berhubungan dan berfungsi secara bersama-sama sesuai tugas masing-masing untuk mencapai tujuan yang telah ditetapkan. Menurut Mc Leod (2005), sistem penjualan adalah suatu proses yang saling mendukung dalam usahanya untuk memenuhi kebutuhan pembeli dan bersama-sama mendapatkan kepuasan dan keuntungan.

Berikut contoh beberapa unsur atau bagian dalam sistem penjualan barang pada suatu perusahaan:

1. Bagian Penjualan
2. Bagian Gudang
3. Bagian Produksi

Untuk elemen atau bagian dalam sistem penjualan dari masing perusahaan mungkin tidak akan sama. Hal ini disebabkan karena kebutuhan informasi yang berbeda-beda antara perusahaan yang satu dengan yang lainnya.

Tujuan sistem penjualan adalah:

1. Mencatat dan mengkonfirmasi order penjualan dengan cepat dan akurat.
2. Memastikan bahwa perusahaan menjual kepada konsumen yang memang layak menerima kredit (sehingga tidak ada kredit macet).
3. Memastikan bahwa konsumen menerima kiriman produk dan jasa tepat waktu, sesuai yang dijanjikan.
4. Menagih tepat waktu dan akurat, sehingga perputaran kas lebih cepat.
5. Mencatat dan mengelompokkan transaksi keuangan secara cepat dan akurat (ke dalam jurnal maupun ke buku besar).
6. Memastikan keamanan aset perusahaan (barang dagangan maupun kas dari penjualan).

2.3 Pengertian Suku Cadang

Suku cadang atau yang disebut *sparepart* biasanya tidak selalu tersedia secara siap ada dipasaran melainkan sangat terbatas keberadaanya. Suku cadang ini merupakan alat penunjang mesin-mesin yang di gunakan untuk memproduksi suatu produk sehingga suku cadang mempunyai peranan yang sangat vital bagi kelangsungan proses produksi disetiap perusahaan manufaktur.

Definisi Suku Cadang (*Sparepart*) Menurut Indrajit, dkk. (2006), dalam bukunya manajemen persediaan menyatakan definisi suku cadang adalah sebagai berikut: “Suku cadang atau *sparepart* adalah suatu alat yang mendukung pengadaan barang untuk keperluan peralatan yang digunakan dalam proses

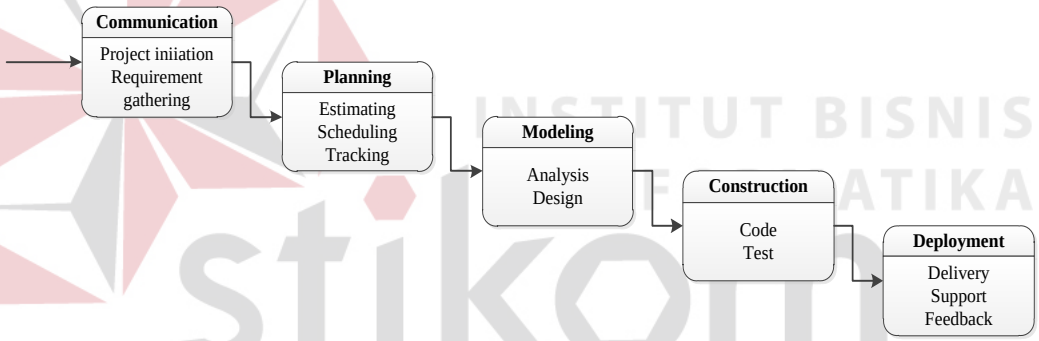
produksi". Berdasarkan definisi diatas, suku cadang merupakan faktor utama yang menentukan jalannya proses produksi dalam suatu perusahaan. Sehingga dapat dikatakan suku cadang ini mempunyai peranan yang cukup besar dalam serangkaian aktivitas perusahaan.

Klasifikasi Suku Cadang (*Sparepart*) Menurut penggunaanya, suku cadang dapat dibagi menjadi tiga jenis. Menurut Indrajit, dkk. (2006), mengklasifikasikan suku cadang ke dalam beberapa jenis yaitu:

1. Suku cadang habis pakai (*consumable parts*) Suku cadang jenis ini adalah suku cadang untuk pemakaian biasa, yaitu yang akan aus dan rusak, kerusakan suku cadang ini dapat terjadi sewaktu-waktu. Oleh karena itu, pengaturan persediaannya haruslah sedemikian rupa sehingga sewaktu-waktu diperlukan haruslah selalu tersedia, atau dapat diadakan dalam waktu singkat sehingga tidak mengganggu jalannya peralatan.
2. Suku cadang pengganti (*replacement parts*) Suku cadang jenis ini adalah suku cadang yang penggantiannya biasanya dilakukan pada waktu *overhaul*, yaitu pada waktu diadakan perbaikan besar-besaran. Waktu *overhaul* ini biasanya dapat dijadwalkan sesuai dengan rekomondasi pabrik pembuat peralatan tersebut. Selain waktu *overhaul* yang dapat dijadwalkan, suku cadang yang perlu diganti dapat juga diperkirakan dengan cukup akurat. Oleh karena itu, biasanya jenis suku cadang ini tidak disimpan dalam persediaan, kecuali untuk peralatan vital.
3. Suku cadang jaminan (*insurance parts*) Suku cadang jenis ini adalah suku cadang yang biasanya tidak pernah rusak, tetapi dapat rusak, dan apabila rusak dapat menghentikan operasi dan produksi. Suku cadang jaminan ini biasanya bentuknya besar, harganya mahal, dan waktu pembuatannya lama.

2.4 Metode Pengembangan *SDLC* (*Systems Development Life Cycle*)

Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun dan kadang dinamakan siklus hidup klasik (*classic life cyle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modelling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.



Gambar 2.1 Model pengembangan *Waterfall* (Pressman, 2015)

Gambar 2.1 menunjukkan tahapan umum dari model proses *waterfall.* Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering.* Model ini melakukan pendekatan secara sistematis dan urut mulai

dari level kebutuhan sistem lalu menuju ke tahap *Communication, Planning, Modeling, Construction, dan Deployment.*

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015) :

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

Construction merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan

dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.5 Kebutuhan Perangkat Lunak

Untuk menentukan kebutuhan perangkat lunak, yang pertama perlu harus diperhatikan setelah definisi dari kebutuhan perangkat lunak, adalah jenis dari kebutuhan tersebut seperti apakah produk atau proses. Keseluruhan proses tersebut dapat menjelaskan perbedaan antara kebutuhan sistem dan perangkat lunak (Jogiyanto, 2008).

2.6 Analisis dan Desain Perangkat Lunak

Analisis sistem atau perangkat lunak dilakukan dengan tujuan untuk dapat mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya.

Perancangan desain perangkat lunak merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun aplikasi (Jogiyanto, 2008).

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya, dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut:

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut, tahap ini disebut desain sistem atau perangkat lunak.

2.7 Konstruksi Perangkat Lunak

Pada tahap ini ialah melakukan konversi hasil desain ke sistem informasi yang lengkap melalui tahapan *coding* atau pengkodean termasuk bagaimana, membuat basis data dan menyiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengodean pengompilasian, memperbaiki dan membersihkan program serta melakukan peminjaman pengujian. *Construction* ini memiliki beberapa tahapan secara umum.

Desain implementasi yang digunakan, bahasa pemrograman yang digunakan, kualitas dari implementasi yang dilakukan, proses pengetesan dan integritas, dalam proses pengimplementasian penelitian ini, digunakan bahasa pemrograman dan beberapa aplikasi pendukung yaitu:

a. Microsoft Visual Studio

Visual Basic .NET 2010 adalah salah satu bahasa pemrograman yang tergabung dalam Microsoft Visual Studio 2010. Visual Studio 2010 yang sering juga disebut dengan VB .Net 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tools*) untuk menghasilkan program-progam aplikasi berbasiskan windows/*desktop*. Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasiskan windows.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol ActiveX, *file Help*, aplikasi Internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program berakhiran EXE yang bersifat *executable* atau dapat langsung dijalankan.

Bahasa Visual Studio cukup sederhana dan menggunakan kata-kata bahasa Inggris yang umum digunakan. Jadi kita tidak perlu lagi menghafalkan sintaks-sintaks maupun format-format bahasa yang bermacam-macam, di dalam Visual Basic semuanya sudah disediakan dalam pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual memudahkan kita untuk mengembangkan aplikasi berbasiskan Windows, bersifat *mouse-driven* (digerakkan dengan *mouse*) dan berdaya guna tinggi (Darmayuda, 2009).

b. Sybase Power Designer

Sybase Power Designer adalah sebuah software pemodelan yang memiliki banyak fungsi diantaranya adalah untuk merancang serta memanage database.Sangat cocok untuk *database* yang berukuran besar serta memiliki tingkat kompleksitas yang cukup rumit.

Sybase Power Designer memiliki beberapa keuntungan sebagai berikut jika dibandingkan dengan menggunakan cara manual atau menggunakan aplikasi pembantu dari vendor, yaitu:

1. Desain *database* mayoritas menggunakan interface berupa tampilan grafik, hal ini berarti orang yang tidak mengerti bahasa SQL juga bisa menggunakan software ini untuk membuat *database* dengan berbagai macam tingkat kesulitan.
2. *Database* dapat diciptakan independen dari RDBMS, Sybase tidak bergantung pada *vendor* tertentu saja untuk mengimplementasikan *design database* yang telah dibuat pada RDBMS seperti MySQL, Oracle, atau Microsoft SQL Server. Sybase dapat mengkonstruksikan *database* yang telah kita buat dalam berbagai macam RDBMS, bahkan RDBMS yang jarang kita dengar juga didukung oleh Sybase.
3. Migrasi *database* menjadi mudah, hal ini dikarenakan Sybase mempunyai fitur untuk mengkoneksikan diri dengan berbagai macam RDBMS seperti Oracle, MySQL, dan Microsoft SQL Server untuk mengkonstruksikan *database* yang telah didesain pada Sybase.
4. Hampir tidak memerlukan pengetahuan mengenai bahasa SQL, semuanya dilakukan dengan klik sana sini di Sybase, namun begitu kita dapat melihat hasil *output* SQL dari design database yang sudah kita buat

Sybase Power Designer cocok untuk digunakan pada saat kita membutuhkan design database yang kuat dan fleksibel dan membutuhkan waktu yang cepat untuk konstruksi database.

c. Microsoft SQL Server

SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa *Query* utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar. Adapun perkembangan SQL Server dimulai dari SQL Server 2000, SQL Server 2005, SQL Server 2008.

Pada tahap ini penulis menggunakan Microsoft SQL Server 2008 karena selain menjadi pendukung pembuatan *database* dari aplikasi pencatatan penjualan suku cadang dan jasa *service*. Microsoft SQL Server 2008 juga mempunyai keunggulan sebagai *failover, clustering, mirroring database,* pengiriman *log* atau replikasi serta sudah terdapat kompresi dan enkripsi data transparan, sehingga kita tidak perlu lagi untuk memodifikasi program dalam mengenkripsi data. SQL Server 2008 juga memiliki sistem kontrol akses yang lebih efisien. SQL Server 2008 ini sendiri juga terintegrasi dengan beberapa produk Microsoft.

2.8 Uji Coba Perangkat Lunak

Uji coba perangkat lunak meliputi verifikasi yang dinamis dari tingkah laku sebuah perangkat lunak yang diwakili oleh beberapa contoh kasus uji coba. Kasus uji coba tersebut dilakukan dengan memberikan masukan kepada perangkat lunak agar muncul tingkah laku/reaksi yang diharapkan, begitu pula sebaliknya. Untuk uji coba perangkat lunak, yang pertama kali diperhatikan adalah fundamental dari uji coba perangkat lunak tersebut. Di dalamnya dijelaskan

mengenai terminologi dari uji coba terkait, kunci masalah dari uji coba, dan hubungan uji coba tersebut dengan aktifitas lainnya di dalam perangkat lunak tersebut. Kedua, yang perlu diperhatikan adalah tingkatan dari uji coba. Di dalamnya dijelaskan tentang target dari uji coba dan tujuan dari uji coba tersebut. Ketiga, yang perlu diperhatikan adalah teknik dari uji coba. Di dalamnya meliputi uji coba berdasarkan intuisi dan pengalaman dari seorang *tester*, diikuti oleh teknik berdasarkan spesifikasi, teknik berdasarkan kode, teknik berdasarkan kesalahan, teknik berdasarkan penggunaan, dan teknik dasar yang relatif tergantung dari aplikasi tersebut. Keempat, yang perlu diperhatikan adalah pengukuran dari uji coba terkait. Di dalamnya dijelaskan bahwa pengukuran tersebut dikelompokkan menjadi dua, yakni yang berhubungan dengan evaluasi ketika uji coba dilakukan serta ketika uji coba selesai dilakukan. Kelima, yang perlu diperhatikan adalah proses uji coba itu sendiri, yang berisi tentang pertimbangan praktis dan aktifitas uji coba.

Untuk penelitian ini, akan dilakukan pengujian *black box* terhadap perangkat lunak yang dibuat. Menurut Pressman (2002: 532), pengujian *black box* adalah proses eksekusi suatu program dengan maksud menemukan kesalahan. Menurut Pressman (2002: 577), teknik pengujian *black box* adalah yang paling lazim selama integrasi. Pengujian *black box* digunakan untuk memperlihatkan bahwa fungsi – fungsi perangkat lunak adalah operasional bahwa *input* diterima dengan baik dan *output* dihasilkan dengan tepat.