

Pengembangan Sistem Informasi Penerimaan Jasa Servis di Bengkel Motor dengan Model Prototype

Tri Retnasari¹, Eva Rahmawati², Cahyo Pambudi³

^{1,2} Program studi Teknik Informatika, STMIK Nusa Mandiri Jakarta

³ Program studi Komputerisasi Akuntansi, AMIK Bina Sarana Informatika Jakarta

¹retna3sari@gmail.com

²pambudic@rocketmail.com

³eva_rijal@gmail.com

Abstrak- Pertumbuhan sepeda motor dari tahun ketahun naik secara signifikan, hal ini berbanding lurus dengan munculnya bengkel-bengkel sepeda motor untuk memenuhi kebutuhan konsumen memperbaiki atau merawat kendaraan yang dimilikinya agar berjalan dengan baik. Kenaikan jumlah sepeda motor di Indonesia ini menimbulkan peluang besar bagi pengusaha untuk membuka usaha dibidang *service* sepeda motor, mulai dari bengkel skala kecil hingga besar pun bermunculan. Namun dalam memudahkan berjalannya usaha dibidang *service* sepeda motor ini, hanya bengkel-bengkel skala besar atau bengkel dengan status Agen Tunggal Pemegang Merek sajalah yang menerapkan sistem terkomputerisasi. Masih banyak bidang usaha *service* sepeda motor ini yang belum menggunakan sistem terkomputerisasi untuk mendukung berjalannya usaha. Berdasarkan hal tersebut, penulis membuat suatu sistem informasi penerimaan jasa *service* sepeda motor yang dapat membantu bagi organisasi dan perusahaan kecil. Sistem ini memungkinkan pengguna secara dengan cepat memahami cara penggunaan sistem terkomputerisasi untuk mendukung proses penerimaan jasa *service* dikarenakan tampilan yang digunakan bersifat *user friendly*. Model pengembangan sistem yang digunakan dalam perancangan sistem ini menggunakan model *Unified Modeling Language* (UML), serta sistem ini mendukung pengolahan data master, transaksi hingga menampilkan laporan yang dibutuhkan, serta sistem ini mendukung adanya hak akses dan perekaman siapa saja yang menggunakan sistem. Aplikasi ini dirancang dengan menggunakan metode *prototype* yang bertujuan agar proses perancangan dah hasil akhir aplikasi yang dibuat dapat digunakan sesuai dengan kebutuhan Bengkel.

Kata Kunci- Model Prototype, UML, Sistem Informasi.

1. PENDAHULUAN

Semakin berkembangnya zaman, pertumbuhan kendaraan pribadi khususnya sepeda motor sangat pesat. Hal ini berbanding lurus dengan bertumbuhnya penyedia jasa servis kendaraan. Namun masuknya sistem teknologi informasi ke Indonesia belum dimanfaatkan sepenuhnya oleh para penyedia jasa servis sepeda yang kurang aman. Dalam perkembangannya, komputer tidak hanya digunakan untuk memproses, menyimpan dan mengolah data menjadi informasi, namun komputer juga dituntut untuk bisa memecahkan suatu masalah serta membantu menyelesaikan pekerjaan lainnya yang diinginkan oleh pengguna secara cepat, tepat dan akurat. Tak hanya itu, Setiap perusahaan mencoba untuk menerapkan sistem informasi agar dapat meningkatkan efisiensi dan efektifitas dalam proses bisnis, sehingga berinvestasi dalam teknologi informasi sangatlah penting untuk memberikan kekuatan bagi perusahaan agar dapat bersaing dan dapat mendukung perusahaan dalam mengambil keputusan [2].

motor khusus skala menengah untuk mendukung proses bisnis yang terjadi. Hal ini disebabkan karena masih kurangnya pengetahuan pegawai mengenal sistem terkomputerisasi. Dilihat dari sisi positif, dengan menggunakan sistem terkomputerisasi, perusahaan dapat melakukan pengelolaan data secara lebih cepat, tepat dan akurat. Sejak komputer ditemukan dan program aplikasi dibuat oleh banyak vendor, muncullah berbagai macam bentuk program aplikasi. Pada generasi awal dikenal program aplikasi berbasis desktop (*desktop based*), lalu diawal tahun 90-an [1]. Penerapan teknologi informasi dalam bidang bisnis merupakan salah satu contoh untuk memenuhi akan kebutuhan informasi yang demikian besar terlebih lagi teknologi ini juga mempunyai nilai akurasi data yang tinggi dalam menunjang keputusan bisnis sehingga pekerjaan akan jauh lebih efektif dan efisien. Teknologi informasi tidak hanya terbatas pada teknologi komputer (perangkat keras dan perangkat lunak) yang akan digunakan untuk memproses dan menyimpan informasi, melainkan juga mencakup teknologi komunikasi untuk mengirim atau menyebarkan informasi pendukung keputusan dan sistem informasi.

Sebagai bukti dalam bidang otomotif roda dua, banyaknya bengkel-bengkel motor yang menjamur di Indonesia khususnya di Jakarta dapat dipastikan hanya bengkel skala besar atau bengkel dengan status Agen Tunggal Pemegang Merek (ATPM) yang menggunakan sistem terkomputerisasi dalam mengelola penerimaan jasa servis dari usaha yang dijalankan. Sedangkan bengkel skala menengah yang masih melakukan pengolahan data penerimaan jasa servis secara manual mempunyai banyak kelemahan, selain membutuhkan waktu dan tenaga yang lebih, keakuratannya pun juga kurang dapat diterima karena kemungkinan kesalahannya sangat besar dan hal yang paling penting adalah penyimpanan data

Pada penelitian sebelumnya yaitu membahas kegiatan operasional bengkel baik dalam pelayanan transaksi dan pengolahan laporan keuangan menggunakan Metode Rapid Application Development (RAD). Adapun tahapan metode RAD meliputi investigasi awal, analisis masalah, analisis kebutuhan sistem, analisis cost benefit, pembuatan database dan perancangan prototype [3].

Dengan adanya sistem penerimaan jasa diharapkan membantu organisasi dan perusahaan kecil bengkel secara dengan cepat memahami cara penggunaan sistem terkomputerisasi untuk mendukung proses penerimaan jasa *service* dikarenakan tampilan yang digunakan bersifat *user*

dalam kewajiban-kewajiban selama suatu periode akuntansi, terutama yang berasal dari aktiva operasi”.

E. Jurnal Penerimaan Kas

Menurut [12] menerangkan bahwa “Jurnal Penerimaan Kas adalah jurnal yang disediakan khusus untuk transaksi penerimaan kas”. Penerimaan kas pada suatu perusahaan dikelompokkan menjadi tiga golongan yaitu:

1. Penerimaan kas dari penjualan tunai
2. Penerimaan kas dari para debitur yang membayar kewajibannya
3. Penerimaan kas dari sumber-sumber lain

Contoh dari jurnal penerimaan kas sebagai berikut :

- | | | |
|------------|-----|-----|
| Kas | xxx | |
| Pendapatan | | xxx |

F. Unified Modeling Language (UML)

- Menurut [13] menerangkan bahwa “UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak”.

Menurut [13], pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:

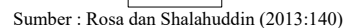


Diagram *Unified Modelling Language* (UML)

- ### G. Use Case Diagram

Menurut [13] menjelaskan bahwa “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

H. Activity Diagram

Menurut [13] mengatakan bahwa “Diagram aktivitas atau *activity diagram* menggunakan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

I. Entity Relationship Diagram (ERD)

Menurut [13] mengatakan bahwa “*Entity Relationship Diagram (ERD)* dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan

D. Pengertian Penerimaan Kas

Menurut [11] menerangkan bahwa penerimaan adalah “Peningkatan dalam aktiva suatu organisasi atau penurunan

basis data relasional”. Berikut adalah komponen dari *Entity Relationship Diagram* (ERD):

1. Entitas (*Entity*)
Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data.
2. Relasi (*Relationship*)
Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya : proses penerimaan jasa servis.
3. Atribut (*Attribute*)
Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data dan *field*. Misalnya nama, alamat, nomor pegawai, gaji adalah atribut dari entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenalan.

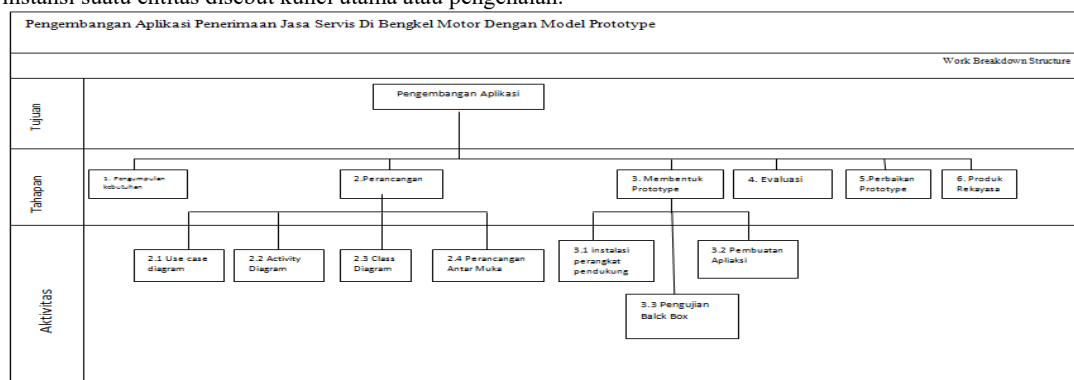
4. Kardinalitas
Pemetaan kardinalitas menyatakan jumlah entitas dimana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi.

J. Logical Record Structure (LRS)

Logical Record Structure adalah ”Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola atau aturan permodelan tertentu dalam kaitanya dengan konvensi ke LRS”.

III. METODE PENELITIAN

Dalam penelitian ini, dilakukan beberapa tahapan untuk mencapai tujuan yang direncanakan.



Gambar 2. Work Breakdown Structure Penelitian

Tahap pengumpulan kebutuhan dan perbaikan merupakan proses melakukan pengumpulan bahan atau data yang sesuai dengan penelitian. Dalam Proses pengumpulan data dilakukan studi literatur terhadap aplikasi yang serupa terhadap penelitian sebelumnya.

Tahapan perancangan dan pembentukan prototype penulis gunakan, yaitu:

a. Analisa kebutuhan *software*

Dalam tahap ini, dilakukan analisa permasalahan terlebih dahulu yang kemudian dikembangkan menjadi suatu logika urutan proses untuk menyelesaikan suatu masalah. Dengan menganalisa kebutuhan *software*, seorang *programmer* dapat menentukan spesifikasi dari fungsi, kemampuan dan fasilitas dari program yang akan dibuat sehingga bermanfaat untuk evaluasi setelah pembuatan program. Model pengembangan sistem yang digunakan dalam perancangan sistem ini menggunakan model *Unified Modeling Language* (UML).

b. Desain

Dalam pembuatan penulisan ini, penulis menggunakan satu *database* MySQL yang berisi 8 tabel yang terdiri dari tabel *cutomer*, pegawai, *technician*, jenis *service*, *sparepart*, transaksi, detail transaksi dan riwayat *login*. Sedangkan *user interface* dibuat dengan memperhatikan kemudahan dalam pengoperasiannya karena biasanya *user interface* dibuat dalam bentuk dialog yang saling berhubungan antara menu, tombol dan layar. *User interface* yang penulis buat yaitu Form Menu

Utama, Form *Customer*, Form Jenis *Service*, Form *Technician*, Form Pegawai, Form *Sparepart*, Form Transaksi, Form Pemilihan Data *Customer*, Form Pemilihan Data Jenis *Service*, Form Pemilihan Data *Technician*, Form Pemilihan Data *Sparepart*, Form Cetak Laporan Harian, Form Cetak Laporan Bulanan, Form Cetak Laporan Tahunan, Form Cetak Riwayat Pengguna, Form *Login*, Form Ganti *Password*, Form *About Programmer*, Form *Backup* dan Form *Restore*.

c. Code Generation

Dalam pembuatan penulisan ini, penulis menggunakan bahasa pemrograman java dan program yang dibuat termasuk jenis pemrograman terstruktur dengan bantuan Netbeans ide 8.1, Xampp control panel v3.2.2, Java Development Kit versi 8.0.730.2 dan iReport untuk membuat laporan-laporan dan faktur yang dibutuhkan sistem ini.

d. Testing

Disini penulis menggunakan *Black Box Testing* yang berarti menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukkan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

e. Support

Ini merupakan tahap terakhir dalam metode pengembangan *software* model *waterfall*. Untuk menjalankan *software* ini dibutuhkan basis data MySQL, Java Development Kit dan sistem operasi yang sudah dicoba adalah Microsoft Windows.

Disamping itu diperlukan spesifikasi minimum *hardware* antara lain Processor Celeron CPU [1005M@1.90GHz](#), RAM dengan kapasitas 2GB dan *Hard Disk* dengan kapasitas 500 GB.

Tahapan evaluasi pelanggan terhadap prototype merupakan tahapan pengujian aplikasi yang sebelumnya dihasilkan dalam tahapan bentuk prototype yang dilakukan oleh pengguna perangkat berbasis Dekstop. Tahapan ini dilakukan dengan melakukan pengujian dan pengisian lembar kuesioner evaluasi kepada pengguna perangkat berbasis Dekstop. Tahapan perbaikan prototype merupakan tahapan dimana dilakukan perbaikan kesalahan atau kelemahan-kelemahan yang terdapat pada aplikasi yang telah dibangun. Tahapan perbaikan prototype dilakukan berdasarkan hasil pengujian yang telah dilakukan sebelumnya oleh pengembang dan responden.

Tahapan produk rekayasa merupakan tahapan dimana produk yang sudah jadi dan siap digunakan oleh pengguna.

IV. HASIL DAN ANALISA

1. Analisa Kebutuhan

Analisis kebutuhan diperlukan untuk mengetahui lebih jelas gambaran dan cara kerja sistem yang ada. Selain itu analisis kebutuhan bertujuan untuk mendefinisikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan suatu perbaikan.

a. Proses *Input*, *Update* dan Hapus Data Master

Data master yang terdapat di *database* program penerimaan jasa *service* adalah tabel *customer*, tabel pegawai, tabel *sparepart*, tabel jenis *service* dan tabel *technician*. Semua data diinput oleh pegawai yang mempunyai hak akses sebagai administrasi, sedangkan pegawai yang mempunyai hak akses sebagai kasir hanya bisa menginput data transaksi penerimaan jasa *service* serta laporan penerimaan jasa *service*. Untuk pegawai yang mempunyai jabatan sebagai *technician* tidak

dapat menggunakan aplikasi penerimaan jasa *service* baik itu merubah data master maupun menginput transaksi yang terjadi. Proses Penginputan Transaksi Penerimaan Jasa *Service* dan Pencetakan Faktur

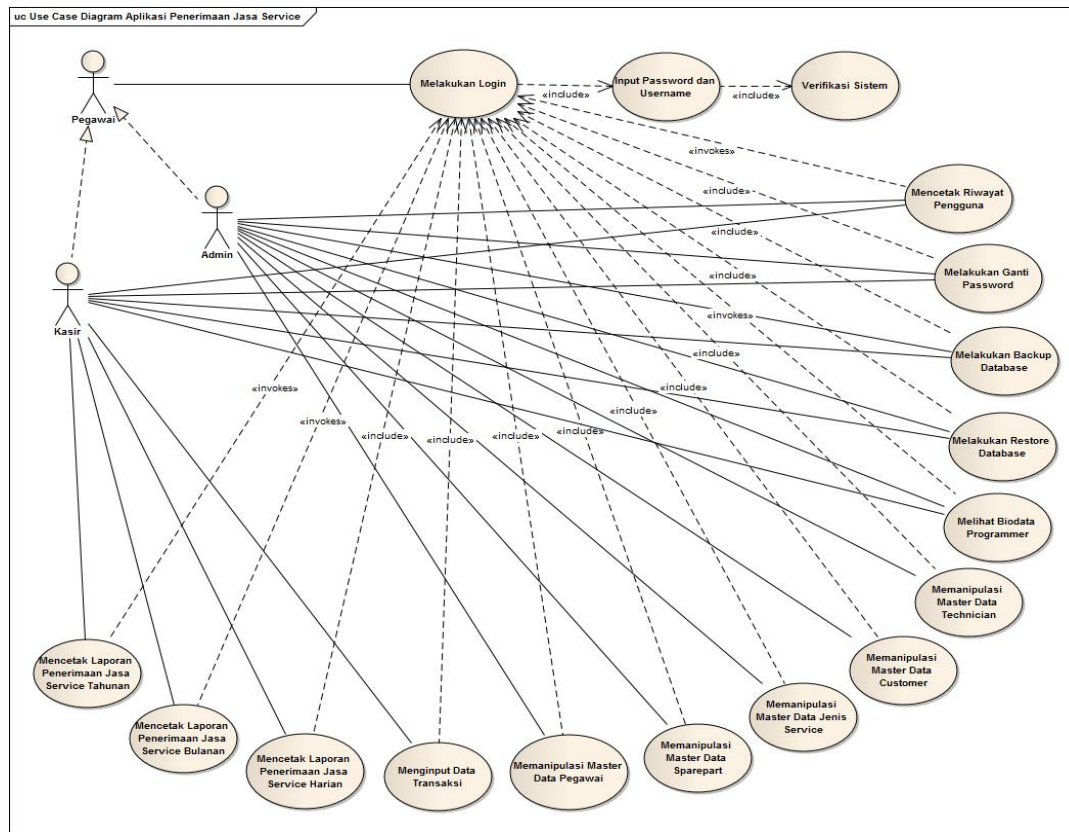
Customer memberikan penjelasan mengenai masalah sepeda motornya kebagian administrasi yang selanjutnya akan diteruskan ke bagian *technician* dan sekaligus menyerahkan Surat Tanda Nomor Kendaraan (STNK) beserta Kartu Member (*Member Card*) kepada bagian administrasi sebagai bukti keaslian kendaraan dan pemeriksaan apakah data *customer* tersebut sudah terdaftar atau belum sebagai member. Jika belum terdaftar, *customer* diberi 2 pilihan apakah ingin menjadi member atau tidak. Jika ingin menjadi member maka bagian administrasi akan membuat data *customer* baru dengan meminta Kartu Tanda Penduduk kepada *customer* sebagai data penginputan *customer*. Jika data *customer* sudah ada atau bisa dibilang sudah terdaftar sebagai member maka *customer* diminta menunggu selama proses *service* sepeda motor berjalan. Jika sepeda motor *customer* telah selesai diservice kemudian bagian kasir akan melakukan penginputan transaksi *service* yang terjadi berdasarkan informasi dari bagian *technician* dan *customer* diminta menuju kasir untuk proses pembayaran. Jika proses pembayaran telah selesai, *customer* diberikan faktur sebagai bukti transaksi *service* yang telah dilakukan.

c. Proses Pencetakan Laporan Penerimaan Jasa *Service*

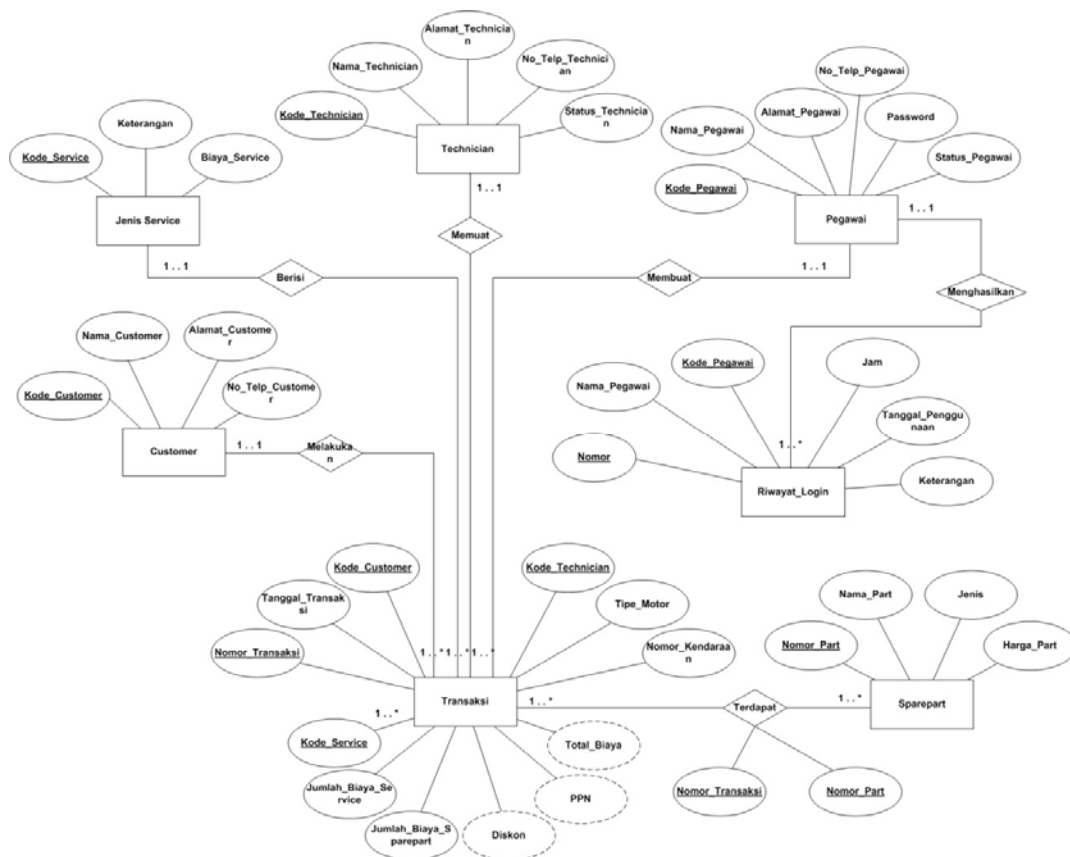
Laporan diambil dari rekapitulasi pada tabel data transaksi dengan tabel detail transaksi. Bagian administrasi maupun bagian kasir dapat melakukan pencetakan laporan penerimaan jasa *service* yang selanjutnya diserahkan kepada pemilik bengkel.

A. Use Case

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat [4].



B. Entity Relation Diagram (ERD)

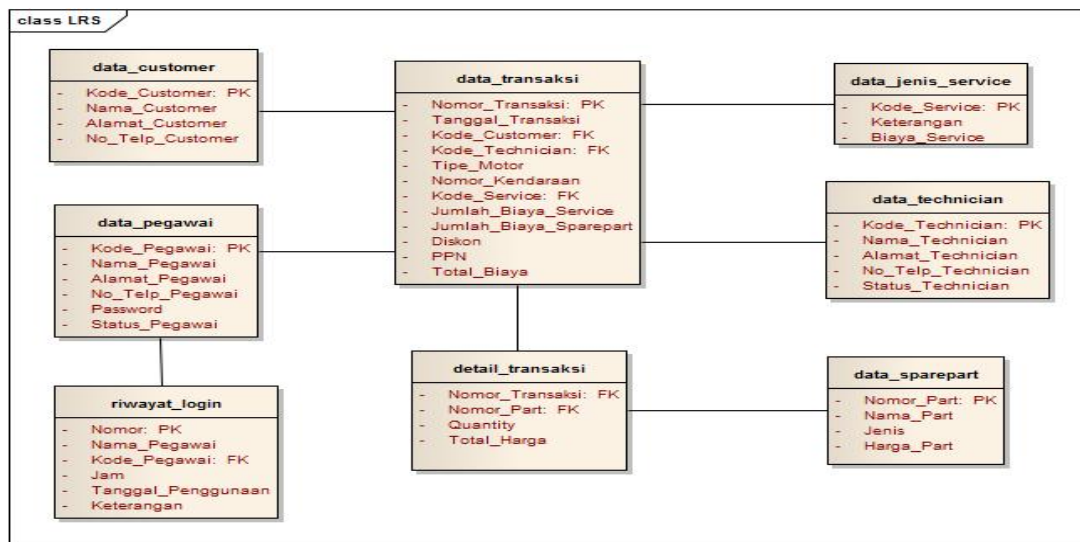


Gambar 4. Entity Relationship Diagram (ERD) Aplikasi Penerimaan Jasa Service

C. Logical Record Structure (LRS)

Logical Record Structure adalah Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti

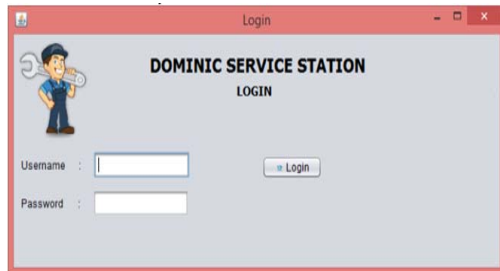
pola atau aturan permodelan tertentu dalam kaitanya dengan konvensi ke LRS [5].



Gambar 5. Logical Record Structure (LRS) Aplikasi Penerimaan Jasa Service

3.2. User Interface

User interface adalah interface yang dibuat untuk user [6]. Biasanya user interface ini dibuat dalam bentuk dialog yang saling berhubungan antara menu, tombol, dan layar.



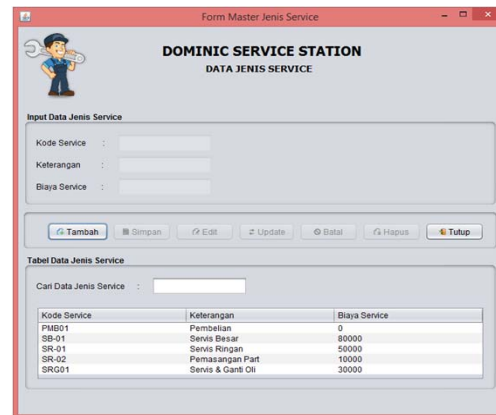
Gambar 6. User Interface Login Aplikasi Penerimaan Jasa Service



Gambar 7. User Interface Tampilan Awal Aplikasi Penerimaan Jasa Service



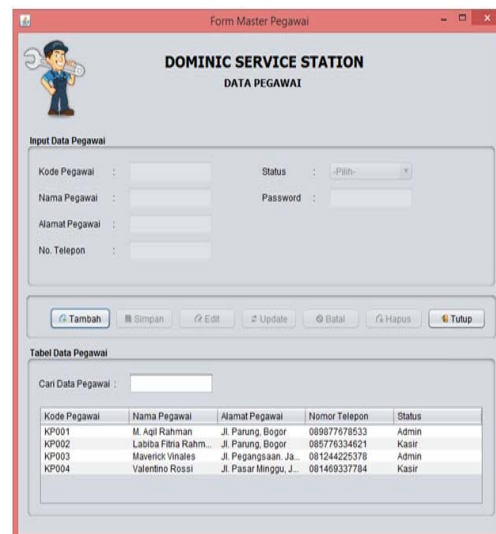
Gambar 8. User Interface Master Customer Aplikasi Penerimaan Jasa Service



Gambar 9. User Interface Master Jenis Service Aplikasi Penerimaan Jasa Service



Gambar 10. User Interface Master Sparepart Aplikasi Penerimaan Jasa Service



Gambar 11. User Interface Master Pegawai Aplikasi Penerimaan Jasa Service

Gambar 12. User Interface Master Technician Aplikasi Penerimaan Jasa Service

Gambar 13. User Interface Transaksi Aplikasi Penerimaan Jasa Service

Gambar 14. User Interface Cetak Laporan Penerimaan Harian Aplikasi Penerimaan Jasa Service

Gambar 17. User Interface Cetak Laporan Penerimaan Bulanan Aplikasi Penerimaan Jasa Service

3.3. Pengujian Black Box Testing

Black box testing atau pengujian kotak hitam yaitu Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [4].

Tabel 1. Hasil Black Box Testing Form Menu Master Jenis Service

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
1	Mengklik tombol Tambah saat Form Master Jenis Service sudah muncul dan kondisi awal tombol Simpan, Edit, Update, Batal dan Hapus dalam kondisi nonaktif		Tombol Tambah berubah menjadi nonaktif dan text field untuk mengisi data aktif atau bisa diinputkan dan tombol Simpan, Batal dan Tutup berubah mejadi aktif	Sesuai harapan	Valid
2	Mengklik tombol Tambah kemudian mengosongkan semua atau beberapa text field penginputan data jenis service dan selanjutnya langsung mengklik	Kode Service : (kosong) Keterangan : (kosong) Biaya Service : (kosong)	Sistem akan menolak atau menunda proses penyimpanan data kedalam database dan menampilkan notifikasi "Lengkapi Data Terlebih Dahulu"	Sesuai harapan	Valid

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
	tombol Simpan				
3	Mengklik salah satu data didalam Tabel Data Jenis <i>Service</i>		Sistem akan menampilkan data yang dipilih kemudian <i>text field</i> akan terisi data	Sesuai harapan	<i>Valid</i>
			yang dipilih tersebut dan tombol Edit, Batal, Hapus dan Tutup dalam keadaan aktif		
4	Memilih data di dalam Tabel Data Jenis <i>Service</i> kemudian mengklik tombol Edit, setelah itu mengklik tombol Update disaat ada salah satu atau lebih data belum terisi	Kode <i>Service</i> : SRG01 Keterangan : (kosong) Biaya <i>Service</i> : 30000	Sistem akan menolak atau menunda proses penyimpanan data kedalam <i>database</i> dan menampilkan notifikasi “Lengkapi Data Terlebih Dahulu”	Sesuai harapan	<i>Valid</i>
5	Mengklik tombol Batal		Sistem akan mengosongkan semua <i>text field</i> pengisian data jenis <i>service</i>	Sesuai harapan	<i>Valid</i>
6	Memilih data di dalam tabel jenis <i>service</i> kemudian mengklik tombol Hapus kemudian memilih pilihan saat notifikasi konfirmasi muncul		Sistem akan menghapus data yang dipilih dari <i>database</i> , sebelumnya akan menampilkan notifikasi konfirmasi “Yakin Data Jenis <i>Service</i> (keterangan) Ingin Dihapus?”. Jika memilih <i>No</i> akan muncul notifikasi “Data Jenis <i>Service</i> (keterangan) Batal Dihapus” namun jika memilih <i>Yes</i> akan muncul notifikasi “Data Jenis <i>Service</i> (keterangan) Berhasil Dihapus”	Sesuai harapan	<i>Valid</i>
7	Mengklik tombol Tutup		Sistem akan mengeluarkan atau menutup Form Master Jenis <i>Service</i>	Sesuai harapan	<i>Valid</i>
8	Menginputkan data pada <i>text field</i> melebihi jumlah batasan karakter di dalam <i>database</i> yang sudah ditentukan	Kode <i>Service</i> : SRG019090 (melebihi batas) Keterangan : <i>Service</i> & ganti oli Biaya <i>Service</i> : 30000	Sistem akan menghentikan penginputan karakter melalui <i>keyboard</i> jika sudah mencapai batasan karakter yang sudah ditentukan di dalam <i>database</i>	Sesuai harapan	<i>Valid</i>
9	Menklik tombol Tambah kemudian dilanjutkan dengan mengisi <i>text field</i> sesuai dengan ketentuan dan langsung mengklik tombol Simpan	Kode <i>Service</i> : SRG01 Keterangan : <i>Service</i> & ganti oli Biaya <i>Service</i> : 30000	Sistem akan menyimpan data yang telah diinputkan ke dalam <i>database</i> kemudian memunculkan notifikasi “Data Jenis <i>Service</i> (keterangan) Berhasil Disimpan”	Sesuai harapan	<i>Valid</i>

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
10	Melakukan proses pencarian data jenis <i>service</i> berdasarkan kode <i>service</i> atau keterangan baik dengan menginputkan secara lengkap atau tidak	Cari Data Jenis <i>Service</i> : SRG	Sistem akan menampilkan hasil pencarian pada tabel data jenis <i>service</i> .	Sesuai harapan	Valid

V. KESIMPULAN dan SARAN

A. Kesimpulan

Dari analisis sampai implementasi sistem maka dapat diambil kesimpulan bahwa pengembangan sistem informasi penerimaan jasa dapat membantu organisasi dan perusahaan kecil bengkel secara cepat dalam proses transaksi. Serta pengguna dapat mudah memahami cara penggunaan sistem terkomputerisasi untuk mendukung proses penerimaan jasa *service* dikarenakan tampilan yang digunakan bersifat *user friendly*. Sistem ini juga mendukung pengolahan data master, transaksi hingga menampilkan laporan yang dibutuhkan, serta sistem ini mendukung adanya hak akses dan perekaman siapa saja yang menggunakan sistem. Dengan menggunakan program aplikasi penerimaan jasa *service* pada bengkel sepeda motor diharapkan dapat mempercepat proses pengolahan data transaksi penerimaan jasa *service* sampai dengan proses pembuatan laporan sehingga waktu yang dibutuhkan untuk mengolah data tersebut relatif lebih singkat. Untuk kedepannya akan dikembangkan fitur-fitur baru yang dapat digunakan oleh pengguna di dalam sistem.

B. Saran

Penulis menyadari bahwa pada Sistem informasi penerimaan Jasa di Bengkel dan tata letak yang diusulkan ini masih memiliki beberapa kekurangan, untuk itu apabila penelitian ini ingin dilanjutkan, penulis ingin memberi beberapa saran yaitu dikembangkan lagi menyesuaikan dengan Teknologi yang lebih *Up To Date*.

Referensi

- [1] M. R. Arief, Pemrograman Web Dinamis menggunakan PHP dan MYSQL, Yogyakarta: Andi Offset, 2011.
- [2] V. D. Mariani, "Analisis Dan Perancangan Sistem Informasi Akuntansi Penjualan, Piutang Dan Penerimaan Kas Pada CV. Dekatama. Seminar Nasional Aplikasi Teknologi Informasi (pp. B-23)," Yogyakarta: Hasugian, H., & Shiddiq, A. N. (2012). *Rancangan Bangun Sistem Informasi Industri Kreatif Bidang Penyewaan Sarana Olahraga*. Semarang: Semantik.
- [3] A. I. J. Putra, "Perancangan Sistem Informasi Berbasis Teknologi Dengan Menggunakan Metode Rapid Application Development (RAD) Pada Bengkel Wijaya Motor," 26 Oktober 2017. [Online]. Available: <http://repository.unika.ac.id/id/eprint/14932>.
- [4] Rosa dan M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika Bandung, 2013.
- [5] H. & S. A. N. Hasugian, "Rancang Bangun Sistem Informasi Industri Kreatif Bidang Penyewaan Sarana Olahraga," dalam *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, Semarang, 2012.
- [6] I. G. H. B. Noerlina, *Perancangan Sistem Informasi Berbasis Object Oriented Studi Kasus*, Jakarta: Mitra Wacana, 2007.
- [7] Munir, R. (2011). *Algoritma dan Pemrograman*. Jakarta: Informatika.
- [8] Yulikuspartono. (2009). *Pengantar Logika dan Algoritma*. Yogyakarta: Andi Offset.
- [9] Yuhefijar. (2008). *Konsep Dasar Sistem Informasi Akuntansi*. Jakarta: Salemba Empat.
- [10] Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Yogyakarta: Andi.
- [11] Hadi, H., & Hastuti, D. (2011). *Kamus Terbaru Ekonomi dan Bisnis*. Surabaya: Reality Publisher.
- [12] Jusup, A. H. (2011). *Dasar-Dasar Akuntansi Jilid 1*. Yogyakarta: Sekolah Tinggi Ilmu Ekonomi YKPN.
- [13] Rosa dan Shalahuddin. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.