

Aplikasi Sistem Informasi Bengkel Sepeda Motor

Hendry Wong^{1,2*}, Linda Wijayanti¹, Markus Tanubrata², Pin Panji Yapinus²

¹Universitas Katolik Indonesia Atma Jaya, Jakarta, Indonesia

²Universitas Kristen Maranatha, Bandung, Indonesia

hendry.w@eng.maranatha.edu*

| Received: 14/05/2025 | Revised: 23/05/2025 | Accepted: 26/05/2025 |

Copyright©2025 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstrak

Dunia usaha cukup banyak permasalahan selain dari penghasilan yang menguntungkan. Sering kali terjadi kesalahan-kesalahan pencatatan data yang menjadi sumber informasi dari dunia usaha tersebut. Begitu juga dalam dunia usaha sebuah bengkel sepeda motor sering kali terjadi kesalahan-kesalahan dalam pencatatan data baik dari proses service atau perbaikan serta penjualan *spare part* dan aksesoris sepeda motor. Dalam usaha bengkel sepeda motor, sering terjadi kesalahan dalam pencatatan persediaan *sparepart* dan aksesoris karena banyaknya jenis *sparepart* dan aksesoris. Selain itu terdapat juga beberapa kesalahan yang perhitungan jumlah total yang harus dibayarkan oleh pelanggan karena masih menggunakan perhitungan secara manual begitu juga dalam pengembalian uang. Melihat dari permasalahan tersebut diperlukan sebuah sistem informasi yang dapat menanggulangi masalah ini, maka akan di bangun sebuah sistem informasi dengan memanfaatkan teknologi informasi baik dari segi perangkat keras maupun perangkat lunak yang mengusung kemudahan penggunaan / *user friendly*. Aplikasi Sistem Informasi Bengkel Sepeda Motor ini akan diimplementasikan dengan MySQL sebagai basis data dan Java Netbeans sebagai perangkat lunak serta menggunakan metoda SDLC dalam rancang bangun sistem informasi. Basis data MySQL dan Java Netbeans merupakan perangkat lunak yang dapat diimplementasikan dalam beberapa sistem operasi seperti Microsoft Windows, MacOS dan Linux sehingga tidak terbatas dalam satu sistem operasi saja. Kesimpulan yang dapat diperoleh dari hasil pembuatan Aplikasi Sistem Informasi Sepeda Motor ini telah memberikan kemudahan kepada karyawan dalam memasukan data baik berupa data *sparepart*, data pelanggan atau transaksi dengan demikian kesalahan perhitungan jumlah total yang harus dibayarkan oleh pelanggan sudah bisa diatasi karena dengan Aplikasi Sistem Informasi Bengkel Sepeda Motor ini semua perhitungan dilakukan secara otomatis sehingga karyawan sudah tidak mengalami kesulitan atau melakukan kesalahan perhitungan jumlah total. Aplikasi ini memberikan kemudahan, dan mengurangi kesalahan-kesalahan yang biasa dilakukan oleh karyawan.

Kata Kunci: MySQL, SDLC, sistem informasi, teknologi informasi, *user friendly*,

Abstract

Business world, there are quite a lot of problems other than profitable income. Often there are errors in recording data that are the source of information from the business world. Likewise in the business world of a motorcycle repair shop, there are often errors in recording data both from the service or repair process and the sale of motorcycle spare parts and accessories. In the motorcycle repair shop business, errors often occur in recording spare part and accessory inventory because of the many types of spare parts and accessories. In addition, there are also several errors in calculating the total amount that must be paid by customers because they still use manual calculations as well as in refunds. Seeing these problems, an information system is needed that can overcome this problem, so an information system will be built by utilizing information technology both in terms of hardware and software that carries ease of use / user friendly. This Motorcycle Repair Shop Information System Application will be implemented with MySQL as a database and Java Netbeans as software and using the SDLC method in designing the information system. MySQL database and Java Netbeans are software that can be implemented in several operating systems such as Microsoft Windows, MacOS and Linux so they are not limited to just one operating system. Conclusions that can be obtained from the result of creating a Motorcycle Information System Application has provided convenience to employees in entering data in the form of spare part data, customer data or transactions so that errors in calculating the total amount to be paid by customers can be overcome because with this Motorcycle Workshop Information System Application all calculations are done automatically so that employees no longer experience difficulties or make errors in calculating the total amount. This application provides convenience and reduces errors that are commonly made by employees.

Keywords: information system, information technology, MySQL, SDLC, user friendly

1. Pendahuluan

Perkembangan teknologi informasi begitu cepat dan teknologi informasi ini sangat mendukung proses bisnis dalam sebuah usaha termasuk usaha dalam bidang bengkel sepeda motor agar proses bisnis bengkel sepeda motor tersebut bisa berjalan lancar. Maka perlu di bangun sebuah aplikasi sistem informasi dengan menggunakan perangkat lunak untuk membangun sebuah aplikasi sistem informasi bengkel sepeda motor untuk menanggulangi permasalahan yang selalu ada dalam usaha bengkel sepeda motor tersebut.

Aplikasi Sistem informasi bengkel sepeda motor memberikan nilai tambah yang memberikan efisiensi pengelolaan informasi dibidang usaha bengkel sepeda motor. Sebuah sistem informasi yang terintegrasi secara digital mempercepat proses administrasi dan pelayanan pelanggan serta mengurangi kesalahan-kesalahan yang bias terjadi baik kesalahan yang disengaja mau tidak disengaja oleh karyawan bengkel motor tersebut termasuk kesalahan dalam perhitungan jumlah total yang harus dibayarkan oleh pelanggan. Memudahkan bagi karyawan untuk mencari harga *sparepart* yang dibutuhkan. Aplikasi sistem informasi bengkel sepeda motor diharapkan memberikan kepuasan kepada karyawan dan juga terutama kepada pelanggan.

Tujuan dari penelitian ini adalah diantaranya. Pertama, Merancang dan membuat sebuah aplikasi sistem informasi bengkel sepeda motor dengan bantuan perangkat lunak yaitu Java Netbeans sebagai bahasa pemrograman untuk membuat aplikasi dan MySQL sebagai *engine database* dengan bahasa SQL. Aplikasi ini dapat di implementasikan di beberapa sistem operasi yaitu windows, macos dan linux. Kedua, Aplikasi sistem informasi bengkel sepeda motor akan menanggulangi permasalahan pencarian serta persediaan *sparepart*. Ketiga, Memberikan efisiensi dalam pengelolaan usaha bengkel sepeda motor. Keempat, Mengurangi kesalahan-kesalahan yang terjadi seperti salah memasukan data dan perhitungan.

2. Metodologi Penelitian

2.1. System Development Life Circle (SDLC)

Metode perancangan dan pembangunan sistem informasi perlu diperhatikan bahwa proses bisnis harus jelas dan terstruktur sehingga diperlukan satu metoda agar perancangan dan pembangunan sistem informasi sesuai dengan proses bisnis. Metoda yang digunakan salah satunya adalah *System Development Life Circle* atau lebih dikenal dengan SDLC.

1. Investigasi

Menyelidiki atau mencari tahu kebutuhan atau kekurangan dari sistem yang ada. Salah satu cara untuk melakukan investigasi adalah dengan melakukan wawancara dengan pelaku sistem agar bisa secara langsung dan efektif mendapatkan informasi tentang kebutuhan sistem yang saat ini berjalan. Wawancara dilakukan dengan terlebih dahulu mempersiapkan pertanyaan-pertanyaan agar pada saat melakukan wawancara sudah ada bahan atau topik yang menjadi sasaran pertanyaan.

Investigasi ini menjadi bagian penting dalam perancangan sistem, sebab dengan melakukan investigasi menjadi jelas alur proses bisnis dari perancangan sistem sampai implementasi.

2. Analisis

Analisa perlu dilakukan agar bisa mendefinisikan kebutuhan sistem serta bagaimana cara untuk menanggulangi kebutuhan atau kekurangan sistem. Dari hasil analisa ini maka kemudian baru dapat dirancang sistem yang baru yang lebih bisa menjawab dari kekurangan atau kebutuhan sistem yang sedang berjalan.

3. Desain

Merancang sistem yang baru dengan merujuk pada investigasi dan analisis dari kebutuhan atau kekurangan sistem tersebut. Rancangan sistem yang diperlukan adalah diantaranya rancangan alur proses antar subsistem atau subproses, kemudian rancangan basis data dan bagian akhir adalah rancangan aplikasi.

4. Implementasi

Implementasi adalah merupakan hasil dari rancangan atau desain sistem baik itu dari alus proses sistem, rancangan basis data, dan racangan aplikasi. Aplikasi Sistem Infromasi Bengkel Sepeda Motor dibangun menggunakan MYSQL sebagai pembuatan basis data, sedangkan untuk pembuatan aplikasi menggunakan bahasa pemrograman Java Netbeans.

5. Perawatan

Memastikan semua prosedur sistem dan aplikasi sudah berjalan dengan baik dan benar. Selain itu bagian alur perawatan juga menjadi bagian dalam menentukan apakah sistem ada pengembangan yang baru.

2.2. Basis Data (*Database*)

Perancangan basis data menjadi bagian yang terpenting dalam pembuatan aplikasi bengkel motor karena merupakan bank data yang dapat diolah untuk menjadi sebuah informasi untuk bengkel motor tersebut.

1. Tabel *supplier*

Tabel tempat untuk menyimpan data *supplier* atau pemasok *sparepart* dari bengkel motor, sehingga bengkel motor mempunyai daftar pemasok untuk memudahkan dalam pencarian dan pembelian *sparepart*. Tabel *supplier* ini memiliki 6 buah kolom dengan satu diantaranya yaitu *kodesupp* adalah merupakan *primary key* dan beberapa kolom dijadikan *not null* yang artinya tidak boleh kosong atau harus diisi data. Tujuan dengan adanya *primary key* adalah agar kolom atau *record* tersebut memiliki sifat yang unik atau terhindar dari duplikasi data. Terdapat 2 buah kolom yang isi datanya adalah berupa angka, tetapi diberikan tipe data *char* yaitu kolom *tlp* dan *hp* dengan tujuan agar bisa diawali dengan angka 0 dan +62. Rancangan tabel *supplier* seperti dibawah ini.

```
Create table supplier(  
Kodesupp char(8) not null primary key,  
Namasupp char(30) not null,  
Kontak char(30) not null,  
Alamat char(150),  
Tlp char(15),  
Hp char(15))
```

2. Tabel *karyawan*

Tabel menyimpan data *karyawan* dari bengkel motor, sehingga bengkel motor mempunyai daftar *karyawan* untuk memudahkan dalam melacak *karyawan* mana yang mengerjakan perbaikan sepeda motor dan juga yang melayani pembeli dalam pembelian *sparepart*. Tabel *karyawan* ini memiliki 5 buah kolom dengan satu diantaranya yaitu *kodekar* adalah merupakan *primary key* dan beberapa kolom dijadikan *not null* yang artinya tidak boleh kosong atau harus diisi data. Tujuan dengan adanya *primary key* adalah agar kolom atau *record* tersebut memiliki sifat yang unik atau terhindar dari duplikasi data. Rancangan tabel *karyawan* seperti dibawah ini.

```
Create table karyawan(  
Kodekar char(8) not null primary key,  
Namakar char(30) not null,  
Alamat char(150),
```

Tlp char(15),
Hp char(15))

3. Tabel *sparepart*

Merupakan tempat untuk menyimpan data *sparepart* dari bengkel motor, sehingga bengkel motor mempunyai daftar *sparepart* untuk memudahkan dalam mencari *sparepart* mana yang diperlukan dalam perbaikan sepeda motor. Tabel *sparepart* ini memiliki 5 buah kolom dengan satu diantaranya adalah merupakan *primary key* dan beberapa kolom dijadikan not null yang artinya tidak boleh kosong atau harus diisi data. Tujuan dengan adanya *primary key* adalah agar kolom atau *record* tersebut memiliki sifat yang unik atau terhindar dari duplikasi data. Rancangan tabel *sparepart* seperti dibawah ini.

```
Create table sparepart(  
Kodepart char(12) not null primary key,  
Namapart char(30) not null,  
Jenispart char(30) not null,  
Jenismotor char(30) not null,  
Hrgbeli numeric(9,0),  
Hrgjual numeric(9,0),  
Ongkos numeric(9,2))
```

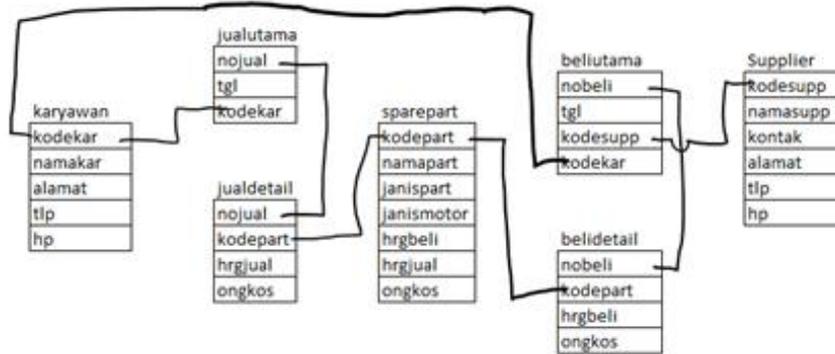
4. Tabel *jualutama* dan tabel *jualdetail*

Kedua tabel ini yaitu tabel *beliutama* dan tabel *belidetil* adalah dua tabel yang menjadi tabel pencatatan data yang merupakan tabel transaksi pembelian *spartpart* kepada pemasok pada saat terjadi kekosongan *spartpart*.

```
Create table jualutama(  
NoJual char(8) not null primary key,  
Tgl date not null,  
Kodekar char(8) not null)  
Create table jualdetail(  
NoJual char(8) not null,  
Kodepart char(12) not null,  
HrgJual numeric(9,0),  
Ongkos numeric(9,2).  
Primary key (nojual,kodepart))
```

5. Relasi basis data

Basis data yang digunakan adalah basis data yang saling berelasi antara tabel satu dengan tabel yang lain. Relasi terbentuk dari tabel satu ke tabel yang lain melalui kolom yang sama dan memiliki kunci. Relasi ini diperlukan juga untuk saling melengkapi antar tabel yang satu dengan tabel yang lain. Relasi basis data bengkel sepeda motor sudah terbentuk seperti gambar dibawah ini.



Sumber: Hendry, dokumentasi pribadi

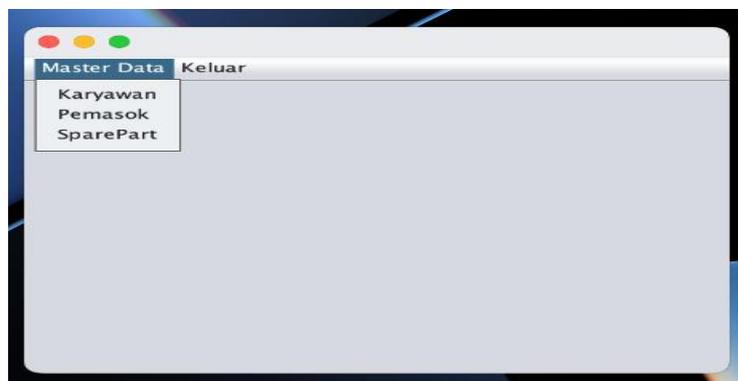
Gambar 1. Relasi Basis Data

2.3. Aplikasi

Setelah pembuatan basis data selesai, maka kemudian dilanjutkan dengan pembuatan aplikasi permodul dari sistem bengkel sepeda motor. Pembuatan aplikasi bengkel sepeda motor ini menggunakan bahasa pemrograman java netbeans versi terbaru saat ini yaitu versi 17. Pemilihan bahasa pemrograman java netbeans agar lebih fleksibel dijalankan diberbagai platform.

1. Form menu utama

Tahap pertama adalah pembuatan modul atau form menu utama yang didalamnya terdapat submenu karyawan, pemasok dan spare part. Modul atau form menu utama ini adalah sebagai tempat berkumpulnya modul-modul atau form yang akan digunakan dalam aplikasi sistem informasi bengkel sepeda motor yaitu form karyawan, pemasok, sparepart, penjualan dan pembelian. Form menu dan form yang lainnya ini dibuat sesederhana mungkin mengikuti aturan user friendly agar memudahkan user untuk mengakases semua form dengan mudah, menghilangkan kebingungan, dan kesalahan-kesalahan yang dapat terjadi.



Sumber: Hendry, dokumentasi pribadi

Gambar 2. Form menu

2. Form karyawan

Modul atau *form* karyawan ini adalah *form* yang fungsinya untuk memasukan data karyawan kedalam tabel karyawan. *Form* ini merupakan antar muka antara pengguna dengan tabel karyawan. Terdapat 4 buah tombol yang masing-masing fungsinya berbeda, yang pertama tombol simpan yang bertugas untuk memvalidasi dan menyimpan data yang diinput pada textbox, tombol hapus berfungsi untuk menghapus data dengan merujuk pada id atau kode karyawan yang bersifat unik sehingga tidak akan terjadi salah penghapusan data, tombol edit berfungsi untuk merubah data apabila ada data yang salah dengan merujuk pada id atau kode karyawan, sedangkan tombol keluar adalah untuk keluar dari *form* karyawan ini. Id atau kode karyawan dalam *form* ini otomatis terbentuk sendiri (*automatic generate*) dengan pola plus 1, sedangkan nama, alamat, telepon, hp merupakan textbox yang harus diisi sendiri secara manual.

- a. *Script SQL* tombol simpan adalah sebagai berikut

```
Insert into karyawan(kodekar, namakar, alamat, telepon, hp)
```

```
Values(textkodekaryawan, textnamakaryawan, textalamat, texttelepon, texthp)
```

- b. *Script SQL* tombol hapus adalah sebagai berikut

```
Delete from karywan where kodekar = textkodekaryawan
```

- c. *Script SQL* tombol edit adalah sebagai berikut

```
Update karyawan set namakar = textnamakaryawan where kodekar = textkodekaryawan
```

- d. Sedangkan untuk tombol keluar hanya berisi *form.close*

No.	Kode Karyawan	Nama Karyaw...	Alamat	Telepon	HP
1	KRR00001	Budi	Sudirman 10	1444	2222

Sumber: Hendry, dokumentasi pribadi

Gambar 3. *Form* karyawan

3. Form pemasok

Modul atau *form* pemasok ini adalah *form* yang fungsinya untuk memasukan data pemasok kedalam tabel pemasok. *Form* ini merupakan antar muka antara pengguna dengan tabel pemasok. Pada *form* pemasok ini juga terdapat 5 buah textbox yang fungsinya adalah menuliskan data sebelum di masukan kedalam tabel pemasok dan 4 buah tombol yang sama dengan 4 buah

tombol yang berada pada *form* karyawan yang berbeda hanya dibagian script sql yang merujuk pada table pemasok begitu juga textbox, yang membedakan 1 buah textbox kontak. Untuk menyimpan data karyawan yang sudah dituliskan pada masing-masing textbox adalah dengan mengakses atau menekan tombol simpan. Script sql masing-masing tombol adalah seperti berikut:

- a. *Script SQL* tombol simpan adalah sebagai berikut

```
Insert into supplier(kodesupp, namasupp, kontak, alamat, telepon, hp)
```

```
Values(textkodesupp, textnamasupp, textkontak, textalamat, texttelepon, texthp)
```

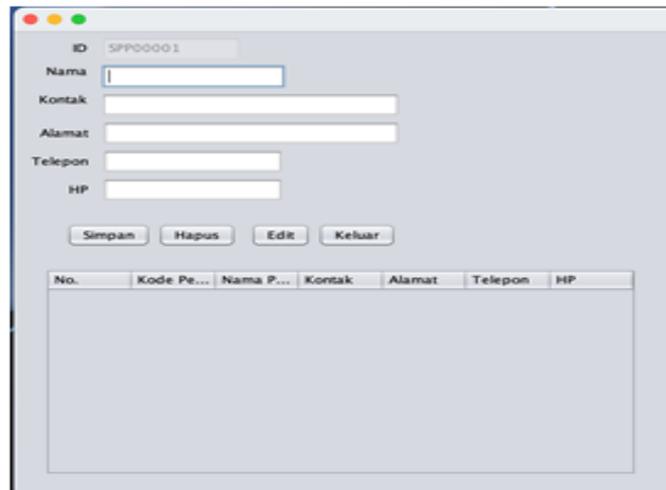
- b. *Script SQL* tombol hapus adalah sebagai berikut

```
Delete from supplier where kodesupp = textkodepemasok
```

- c. *Script SQL* tombol edit adalah sebagai berikut

```
Update supplier set namasupp = textnamapemasok where kodesupp = textkodepemasok
```

- d. Sedangkan untuk tombol keluar hanya berisi *form.close*



No.	Kode Pe...	Nama P...	Kontak	Alamat	Telepon	HP
-----	------------	-----------	--------	--------	---------	----

Sumber: Hendry, dokumentasi pribadi

Gambar 4. *Form* pemasok

4. *Form* sparepart

Modul atau *form spare part* ini adalah *form* yang fungsinya untuk memasukan data *spare part* kedalam tabel *spare part*. *Form* ini merupakan antar muka antara pengguna dengan tabel *spare part*. Pada *form spare part* ini terdapat 6 buah *textbox* yang fungsinya adalah untuk menuliskan data yang nantinya akan dimasukan kedalam table *sparepart*. Yang pertama adalah *textbox* kode part untuk menuliskan data kodepart, *textbox* nama part untuk menuliskan data nama part yang sesuai dengan kodepart, *textbox* jenis part adalah untuk menuliskan jenis part berdasarkan kode part, *textbox* harga jual dan harga beli adalah untuk menuliskan data harga penjualan dan harga pembelian dan *textbox* ongkos untuk menuliskan data ongkos. Setelah semua data ditulis sesuai dengan dengan masing-masing *textbox* kemudian untuk menyimpan data kedalam table *sparepart* dengan mengakses atau menekan tombol simpan. *Script SQL* masing-masing tombol adalah sebagai berikut:

- a. *Script SQL* tombol simpan adalah sebagai berikut

Insert into sparepart(kodepart, namapart, jenispart, jenismotor,hargabeli,hargajual, ongkos)

Values(textkodepart, textnamapart, textjenispart, textjenismotor, texthargabeli, texthargajual, textongkos)

- b. *Script SQL* tombol hapus adalah sebagai berikut

Delete from *sparepart* where kodepart = textkodepart

- c. *Script SQL* tombol edit adalah sebagai berikut

Update sparepart set namapemasok = textnamapemasok where kodepemasok = textkodepemasok

- d. Sedangkan untuk tombol keluar hanya berisi *form.close*

Sumber: Hendry, dokumentasi pribadi

Gambar 5. *Form sparepart*

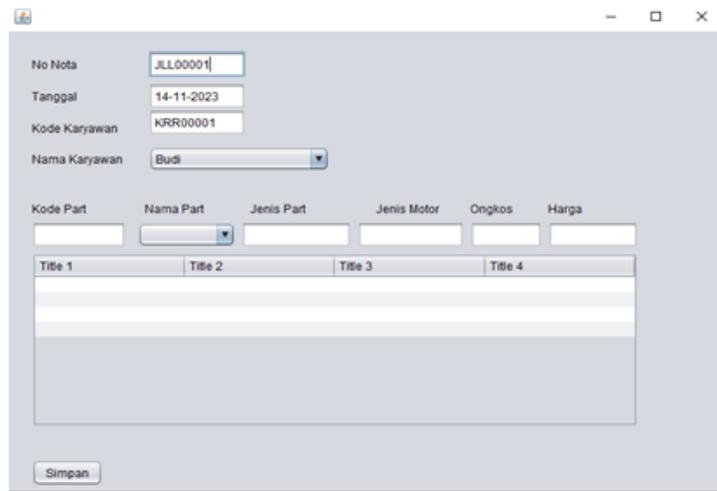
5. *Form* penjualan

Modul atau *form* penjualan ini adalah *form* yang fungsinya untuk memasukan data penjualan kedalam tabel jualutama, table jualdetail, table sparepart dan table karyawan serta semua table ini saling berelasi satu sama lain dengan merujuk pada kolom yang isinya sama dan memiliki kunci yaitu antra *primary key* dan *foreign key*, seperti dapat dilihat pada gambar 1 relasi basis data. Pada *form* penjualan ini tidak terlalu banyak terdapat tombol tetapi hanya ada satu tombol saja yaitu tombol simpan yang fungsinya adalah untuk menyimpan data kedalam table jualdetail. Tujuannya adalah untuk meminimalisir kesalah-kesalahan proses dengan salah mengakses atau menekan tombol. Penghapusan data bisa langsung dengan mengakses grid dibawah dengan menekan mengklik 2 kali dan isi dari table jual detail tertampil di textbox kodepart, textboxpart, textjenispart, textjenismotor, textongkos, txtharga dengan *script sql* sebagai berikut :

Select jualdetail.nojual, jualdetail.kodepart, *sparepart*.namapart, *sparepart*.jenispart, *sparepart*.jenismotor, jualdetail.hrgjual, jualdetail.ongkos *from* jualdetail

Join sparepart on sparepart.kodepart=jualdetail.kodepart

Where kodepart = textkodepar.text



The screenshot shows a web-based sales form. At the top, there are input fields for 'No Nota' (JLL0001), 'Tanggal' (14-11-2023), 'Kode Karyawan' (KRR0001), and 'Nama Karyawan' (Budi). Below these are fields for 'Kode Part', 'Nama Part', 'Jenis Part', 'Jenis Motor', 'Ongkos', and 'Harga'. A table with four columns labeled 'Title 1', 'Title 2', 'Title 3', and 'Title 4' is present. A 'Simpan' button is at the bottom left.

Sumber: Hendry, dokumentasi pribadi

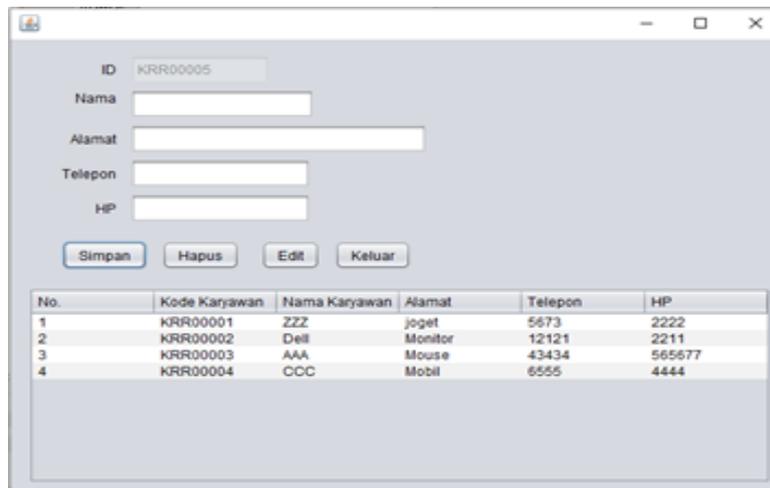
Gambar 6. Form penjualan

3. Hasil dan Pembahasan

Pembuatan Aplikasi Sistem Informasi Bengkel Sepeda Motor telah selesai dan selanjutnya adalah tahap uji coba aplikasi yang telah selesai dibuat

1. Modul karyawan dan tabel karyawan

Hasil dari modul karyawan beserta tabel karyawan dapat dilihat pada gambar dibawah ini



The screenshot shows an employee management interface. It includes a form with fields for 'ID' (KRR0005), 'Nama', 'Alamat', 'Telepon', and 'HP'. Below the form are buttons for 'Simpan', 'Hapus', 'Edit', and 'Keluar'. At the bottom, there is a table with the following data:

No.	Kode Karyawan	Nama Karyawan	Alamat	Telepon	HP
1	KRR0001	ZZZ	joget	5673	2222
2	KRR0002	Dell	Monitor	12121	2211
3	KRR0003	AAA	Mouse	43434	56577
4	KRR0004	CCC	Mobil	6555	4444

Sumber: Hendry, dokumentasi pribadi

Gambar 7. Modul karyawan

Data karyawan diinputkan kedalam tabel karyawan melalui modul karyawan, untuk id karyawan otomatis terbentuk lalu mulai dari nama, alamat, telepon serta hp diinputkan secara manual sesuai dengan data karyawan. Setelah tombol simpan di tekan maka data yang diinputkan akan masuk kedalam tabel karyawan dan ditampilkan dalam data *grid* seperti terlihat pada gambar

7. Apabila terjadi kesalahan *input* atau memasukan data maka bisa dengan cara mengakases tombol simpan dengan terlebih dahulu memilih data yang akan dihapus pada data *grid*. Begitu juga apabila akan memperbaiki data yang sudah disimpan, harus memilih data yang akan dihapus terlebih dahulu pada data *grid*.

Modul karyawan ini berfungsi dengan baik tanpa ada kesalahan dan dapat dilihat pada gambar 8 keberhasilan dari input data karyawan serta pada tabel karyawan yang terdapat 4 *record* data karyawan. Modul dan tabel karyawan ini menjadi salah satu bagian yang penting sehingga memudahkan pemilik bengkel sepeda motor mendapatkan informasi tentang karyawan-karyawan yang bekerja dibengkel sepeda motor tersebut dan juga data karyawan.



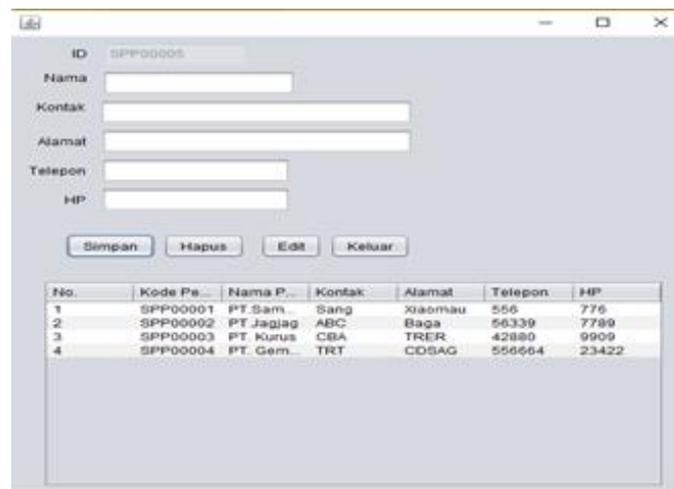
Kodekar	Namakar	Alamat	Tlp	Hp
KRR00001	ZZZ	joget	5673	2222
KRR00002	Dell	Monitor	12121	2211
KRR00003	AAA	Mouse	43434	565677
KRR00004	CCC	Mobil	6555	4444

Sumber: Hendry, dokumentasi pribadi

Gambar 8. Tabel karyawan

2. Modul pemasok dan tabel pemasok

Hasil dari modul pemasok beserta tabel pemasok dapat dilihat pada gambar dibawah ini.



No.	Kode Pe...	Nama P...	Kontak	Alamat	Telepon	HP
1	SPP00001	PT Sam	Sang	xiaomai	555	776
2	SPP00002	PT Jagag	ABC	Baga	56339	7789
3	SPP00003	PT Kurus	CBA	TRER	42880	9909
4	SPP00004	PT Gem	TRT	CDSAG	56664	23422

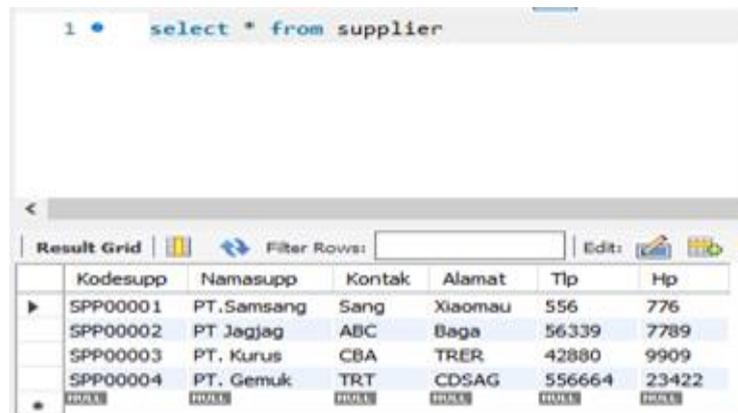
Sumber: Hendry, dokumentasi pribadi

Gambar 9. Modul pemasok

Data pemasok diinputkan kedalam tabel pemasok melalui modul pemasok, untuk id pemasok otomatis terbentuk lalu mulai dari nama, alamat, telepon serta hp diinputkan secara manual sesuai dengan data pemasok. Setelah tombol simpan di tekan makan data yang diinputkan

akan masuk kedalam tabel pemasok dan ditampilkan dalam data grid seperti terlihat pada gambar 9. Apabila terjadi kesalahan *input* atau memasukan data maka bisa dengan cara mengakases tombol simpan dengan terlebih dahulu memilih data yang akan dihapus pada data *grid*. Begitu juga apabila akan memperbaiki data yang sudah disimpan, harus memilih data yang akan dihapus terlebih dahulu pada data *grid*.

Modul pemasok ini sudah berhasil dan berfungsi dengan baik seperti terlihat pada gambar 10, terdapat 4 *record* data pada table pemasok yang menunjukkan keberhasilan dari modul pemasok dalam inpat data tanpa adanya kesalahan. Modul pemasok ini juga merupakan bagian penting pada bengkel sepeda motor ini untuk bisa mendapatkan informasi pembelian *sparepart* sehingga pemilik bengkel bisa langsung mendapatkan informasi tentang pemasok yang sesuai dengan *sparepart* yang dibutuhkan.



The screenshot shows a database query window with the SQL statement "select * from supplier". Below the query, there is a "Result Grid" displaying a table with 7 columns: Kodesupp, Namasupp, Kontak, Alamat, Tlp, and Hp. The table contains 4 records.

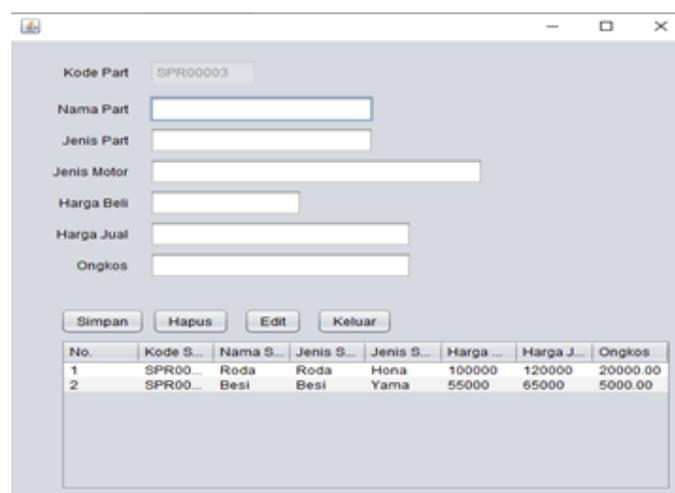
Kodesupp	Namasupp	Kontak	Alamat	Tlp	Hp
SPP00001	PT. Samsang	Sang	Xiaomau	556	776
SPP00002	PT Jagjag	ABC	Baga	56339	7789
SPP00003	PT. Kurus	CBA	TRER	42880	9909
SPP00004	PT. Gemuk	TRT	CDSAG	556664	23422

Sumber: Hendry, dokumentasi pribadi

Gambar 10. Tabel pemasok

3. Modul *sparepart* dan tabel *sparepart*

Hasil dari modul *sparepart* beserta tabel *sparepart* dapat dilihat pada gambar dibawah ini.



The screenshot shows a form for managing spare parts. It includes input fields for "Kode Part" (pre-filled with "SPR00003"), "Nama Part", "Jenis Part", "Jenis Motor", "Harga Beli", "Harga Jual", and "Ongkos". Below the form are buttons for "Simpan", "Hapus", "Edit", and "Keluar". At the bottom, there is a table with 8 columns: No., Kode S..., Nama S..., Jenis S..., Jenis S..., Harga ..., Harga J..., and Ongkos. The table contains 2 records.

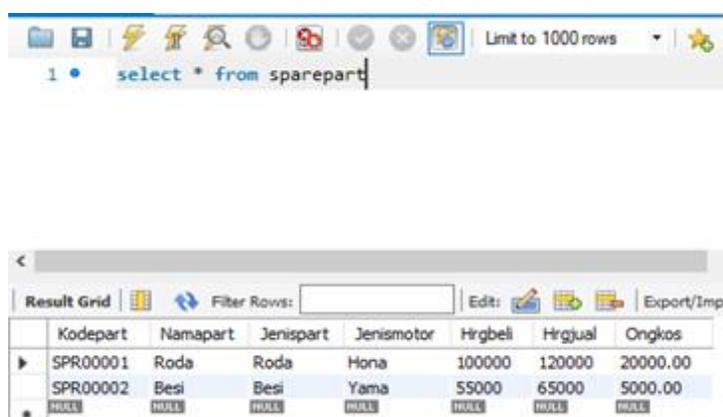
No.	Kode S...	Nama S...	Jenis S...	Jenis S...	Harga ...	Harga J...	Ongkos
1	SPR00...	Roda	Roda	Hona	100000	120000	20000.00
2	SPR00...	Besi	Besi	Yama	55000	65000	5000.00

Sumber: Hendry, dokumentasi pribadi

Gambar 11. Modul *sparepart*

Data *sparepart* diinputkan ke dalam tabel *sparepart* melalui modul *sparepart*, untuk kode part otomatis terbentuk lalu mulai dari nama part, jenis part, jenis motor, harga beli serta harga jual diinputkan secara manual sesuai dengan data *sparepart*. Setelah tombol simpan di tekan maka data yang diinputkan akan masuk kedalam tabel *sparepart* dan ditampilkan dalam data *grid* seperti terlihat pada gambar 11. Apabila terjadi kesalahan *input* atau memasukan data maka bisa dengan cara mengakases tombol simpan dengan terlebih dahulu memilih data yang akan dihapus pada data *grid*. Begitu juga apabila akan memperbaiki data yang sudah disimpan, harus memilih data yang akan dihapus terlebih dahulu pada data *grid*.

Melalui tabel 12 dapat dilihat terdapat 2 record data *sparepart* yang merupakan hasil dari input data melalui modul *sparepart* dan hal ini menunjukkan keberhasilan dari modul *sparepart* dalam penginputan data. Bagian dari modul dan tabel *sparepart* ini penting agar bengkel bisa mendapatkan informasi tentang jenis *sparepart* yang sesuai dengan jenis motor serta harga beli dan harga jual, hal ini akan memudahkan bengkel untuk menentukan jenis *sparepart* yang sesuai.



The image shows a screenshot of a database application interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered in a text box: 'select * from sparepart'. Below the query, there is a 'Result Grid' displaying two rows of data. The columns are labeled: Kodepart, Namapart, Jenispart, Jenismotor, Hrgbeli, Hrgjual, and Ongkos. The first row has values: SPR00001, Roda, Roda, Hona, 100000, 120000, 20000.00. The second row has values: SPR00002, Besi, Besi, Yama, 55000, 65000, 5000.00. Below the grid, there are several 'NULL' labels.

Kodepart	Namapart	Jenispart	Jenismotor	Hrgbeli	Hrgjual	Ongkos
SPR00001	Roda	Roda	Hona	100000	120000	20000.00
SPR00002	Besi	Besi	Yama	55000	65000	5000.00

Sumber: Hendry, dokumentasi pribadi

Gambar 12. Tabel *sparepart*

4. Kesimpulan

Perancangan dan aplikasi sistem informasi bengkel sepeda motor telah selesai dikembangkan dan berdampak pada proses bisnis bengkel yang lebih efisien, pencarian data lebih mudah dan meminimalisir kesalahan-kesalahan yang sering terjadi. Aplikasi ini masih menggunakan bahasa pemrograman tidak berbasis website, sehingga masih terbatas di pc lokal saja. Bahasa Java Netbeans dan engine database MySQL yang menjadi pilar dari aplikasi yang telah dikembangkan dapat dijalankan di beberapa sistem operasi yaitu windows, linux dan macos. Metodu yang digunakan yaitu SDLC memberikan proses perancangan sistem menjadi lebih terstruktur. Kemudian beberapa saran untuk keberlanjutan penelitian adalah, Aplikasi menggunakan bahasa pemrograman berbasis *website* sehingga dapat di jalankan melalui browser sehingga aplikasi dapat diakses dimana pun tanpa dibatasi hanya di satu lokasi saja. Tampilan per modul dibuat lebih *user friendly* agar minim kesalahan dan lebih menarik. Sistem bisa ditambahkan untuk menghitung pendapatan insentif bagi para karyawan bengkel sepeda motor tersebut.

Daftar Pustaka

Ward, J., Peppard, J. (2003). *Strategic Planning for Information Systems*. Jhon Wiley & Sons,

LTD.

- Burlton, R.T. (2001). *Business Process Management: Profiting from Process*. SAMS.
- Dunn, C.L., J., Cherrington, O., Hollander, A. S. (2005). *Enterprise Information Systems: A Pattern-Base Approach*. Mc Graw-Hill.
- Huda, M., Nugroho, B. (2010). *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. Elex Media Komputindo.
- Nugroho, B. (2012). *Panduan Membuat Program Toko dengan PHP, MySQL dan Dreamweaver: Point of Sale*. Alifmedia.
- Kadir, A., Triwahyuni, T.C. (2003). *Pengenalan Teknologi Informasi*. Andi Yogyakarta.
- Nugroho, A. (2004). *Konsep Pengembangan Sistem Basis Data*. Informatika Bandung.
-