

# Rancang Bangun Aplikasi *E-Report* Berbasis *Web* pada PT Bersama Makmur Raharja

**Nilia Natalia, Nur Rohmah Arraudoh Mahmud**

Program Studi Teknik Komputer, Politeknik Sukabumi  
Jl. Babakan Sirna No.25, Benteng, Kec. Warudoyong, Kota Sukabumi, Jawa Barat 43132  
nilianatalia@polteksmi.ac.id

---

---

## Abstrak

Dalam suatu perusahaan atau industri data dan informasi merupakan suatu hal yang penting untuk melakukan suatu proses bisnis karena persaingan dalam dunia industri semakin ketat sehingga perusahaan pun akan selalu berusaha agar terus menjadi yang pertama dalam memuaskan mitra perusahaan agar tidak beralih ke yang lain. Karena itu pihak perusahaan perlu mengembangkan sistem yang belum dapat memenuhi kebutuhan dalam laporan penjualan, mengetahui laporan penjualan setiap saat dan perkembangan penjualan dapat di pantau setiap saat dan dimana saja secara *online*. Maka dengan perancangan website ini dapat mempermudah kinerja para salesman untuk metode input order harian dan membantu admin untuk *report* data order bulannya.

**Kata kunci:** Perancangan *web*, sistem informasi, sales, *report*.

---

---

## I. PENDAHULUAN

Kemajuan ilmu pengetahuan dalam teknologi menjadi bagian kehidupan kita sehari-hari. Hampir semua kalangan masyarakat memiliki telepon seluler, personal digital *assistant* atau sejenisnya yang mereka gunakan untuk berkomunikasi lewat suara, memeriksa email atau manfaat teknologi lainnya.

Dalam proses bisnisnya, PT Bersama Makmur Raharja belum menggunakan teknologi sistem informasi, sehingga terdapat beberapa kendala yang ditemukan seperti pencatatan data transaksi penjualan yang membutuhkan waktu cukup lama karena data diketik secara manual. Dalam penginputan data penjualannya pun seringkali terdapat kesalahan sehingga data penjualan tidak lagi akurat, keterbatasan mempengaruhi ketidak efektifan dalam memberikan laporan, menghitung transaksi sehari-hari karena masih menjadi kebiasaan. Informasi yang ditentukan kemudian dikirim melalui group aplikasi *WhatsApp*. Interaksi ini juga berisi bahwa admin harus memeriksa secara individual sales mana yang telah menyerahkan laporan dan selanjutnya memberikan teguran kepada sales yang belum menyerahkan laporan menjelang akhir jam kerja.

Dengan itu kita diharapkan dapat mengatasi masalah yang ada dengan memakai penyempurnaan inovatif serta kecepatan dan ketepatan dalam memberikan data sehingga dalam menyelesaikan pekerjaan kita akan mendapatkan hasil yang maksimal dengan alasan bahwa dalam suatu organisasi atau industri, informasi dan data adalah sesuatu hal yang penting untuk menindaklanjuti dengan sesuatu proses bisnis.

Berdasarkan permasalahan tersebut sangat diperlukan aplikasi monitoring yang bisa mempermudah untuk menghitung data yang telah di input dan menyimpan laporan bulanan. Maka penulis bermaksud untuk membuat aplikasi berbasis *web* monitoring pada PT. Bersama Makmur Raharja tujuannya untuk membantu sales dalam proses *booking* penjualan menjadi lebih cepat dan mudah dapat dijadikan solusi bagi perusahaan, sales dapat segera menginput data order dari pelanggan dan mengirim data ke perusahaan langsung dari lapangan tanpa harus kembali ke kantor, aktivitas penjualan dapat dilakukan lebih cepat dan fleksibel.

## II. METODE PENELITIAN

Dalam penulisan ini metode yang digunakan merupakan metode *waterfall* dengan tahapan analisa kebutuhan, desain system, penulisan kode

program, pengujian program, dan penerapan program.

1. Analisa Kebutuhan.

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan penelitian dan wawancara.

2. Desain Sistem.

Tahapan dimana dilakukannya penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem.

3. Penulisan Kode Program.

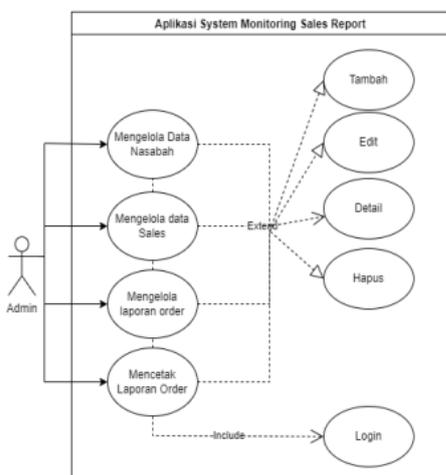
Penulisan kode program atau coding merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem.

4. Pengujian Program.

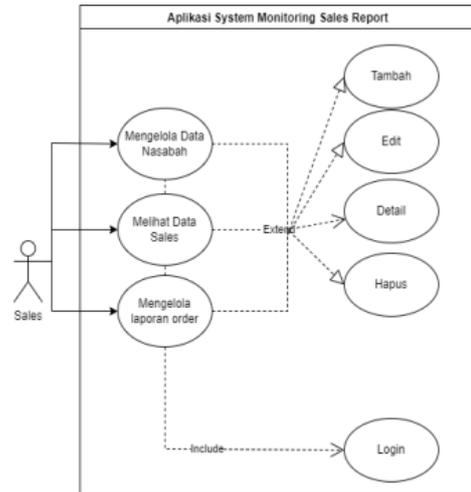
Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik.

5. Penerapan Program.

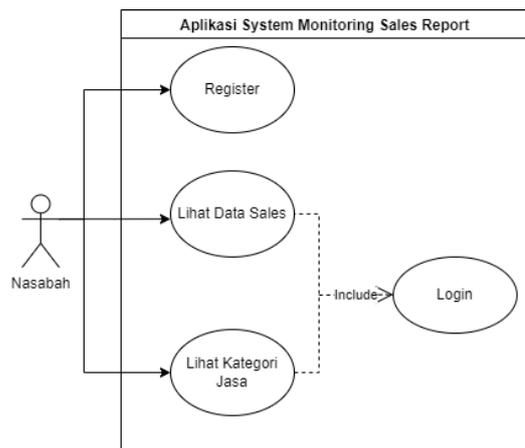
Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan atau karena pelanggan membutuhkan perkembangan fungsional. Berikut use case diagram dari alur aplikasi ditunjukkan pada gambar di bawah ini.



Gambar 1a. Use Case diagram aplikasi

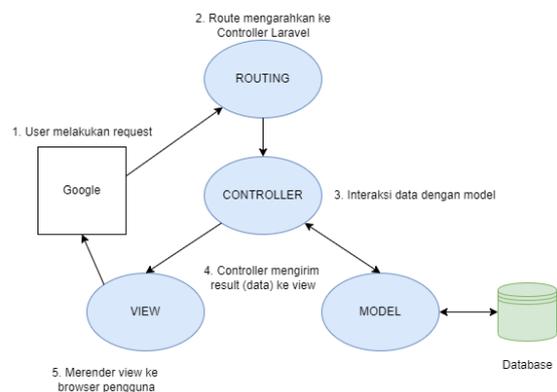


Gambar 1b. Use Case diagram aplikasi



Gambar 1c. Use Case diagram aplikasi

Website yang akan dibuat ini nantinya dapat membantu sales dalam melakukan pencatatan order, dan memberikan informasi data penjualan yang akurat sehingga tidak ada data yang selisih.



Gambar 1. Gambaran Umum Sistem Laravel

Ada 5 konsep arsitektur pada framework laravel yang mempunyai masing-masing fungsi diantaranya:

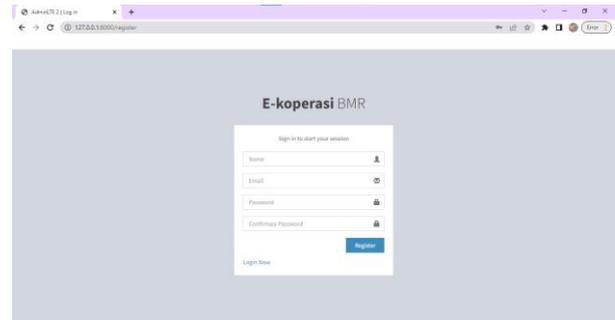
1. *Routes*, berfungsi sebagai pemberi akses pada setiap request sesuai alur yang telah *di* tentukan.
2. *Controller*, adalah bagian yang menjadi penghubung antara model dan *view*. *Controller* memiliki *perintah-perintah* yang berfungsi *untuk* memproses bagaimana data ditampilkan dari Model ke *View* *atau* sebaliknya.
3. Model, merupakan sekumpulan data yang memiliki fungsi-fungsi *untuk* mengelola suatu table pada sebuah *database*. Struktur pemodelan data pada *laravel* yakni memiliki fungsi yang terdiri dari table, *primaryKey* dan fillable. Dimana ketiga fungsi tersebut harus *di* protecte.
4. Pada bagian table harus diisi dengan nama table yang sesuai pada *database*, *di* bagian *primaryKey* harus diisi sesuai primary key pada table tersebut dan pada bagian fillable diisi dengan bagian-bagian yang mencakup dalam table tersebut *d*. *View*, merupakan file yang berisi kode *html* (*HyperText Markup Language*) yang berfungsi *untuk* menampilkan suatu data ke dalam browser. *Format view* pada *laravel* harus menggunakan istilah blade, contohnya seperti: *view.blade.php*.
5. *Migrations*, merupakan proses perancangan suatu table, dalam hal ini migrations berfungsi sebagai blueprint *database* *atau* dapat diistilahkan sebagai penyedia sistem kontrol *untuk* skema *database*.

### III. HASIL DAN PEMBAHASAN

Sesudah melakukan perancangan sistem, selanjutnya kita melakukan implementasi rancangan. Adapun implementasi dilakukan dengan bahasa pemrograman *Framework* Laravel. Hasil implementasi berupa sebuah website. Di sini akan dijelaskan semua antarmuka *activity* yang sudah dibuat oleh penulis dan dapat diakses oleh pengguna selama terdapat koneksi internet.

#### 1. Tampilan Halaman Register

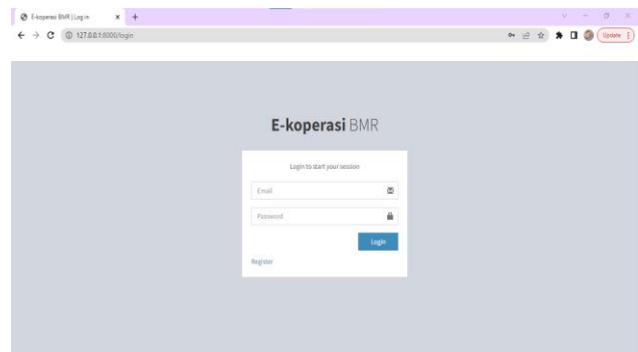
Tampilan halaman *register* merupakan halaman yang menampilkan *form register* *untuk* user melakukan sebuah proses pendaftaran *atau* pembuatan akun. Berikut ini merupakan antarmuka halaman *register*, ditunjukkan pada gambar dibawah ini.



Gambar 3. Tampilan Halaman Register

#### 2. Tampilan Halaman Login

Tampilan halaman *login* merupakan halaman yang menampilkan *form login* *untuk* user melakukan sebuah proses *login*. Berikut ini merupakan antarmuka halaman *login*, ditunjukkan pada gambar dibawah ini.



Gambar 4. Tampilan Halaman Login

#### 3. Tampilan Halaman Admin

Tampilan halaman *admin* merupakan tampilan halaman awal setelah *admin* melakukan proses *login*. Berikut ini merupakan antarmuka yang menampilkan menu-menu apa saja yang tampil *di* halaman *admin*, ditunjukkan pada gambar dibawah ini.



Gambar 5. Tampilan Halaman Admin

#### 4. Tampilan Halaman Sales

Tampilan halaman sales merupakan tampilan halaman awal setelah sales melakukan proses *login*.

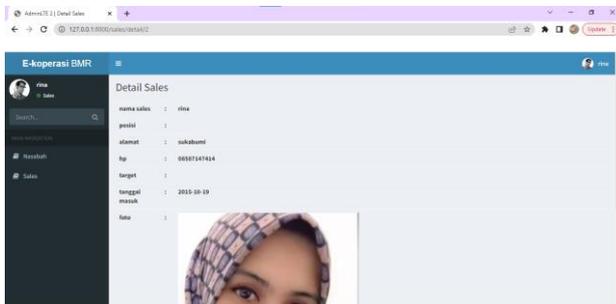
Berikut ini merupakan antarmuka yang menampilkan menu-menu apa saja yang tampil di halaman sales, ditunjukkan pada gambar dibawah ini.



Gambar 6. Tampilan Halaman Sales

### 5. Tampilan Detail Sales

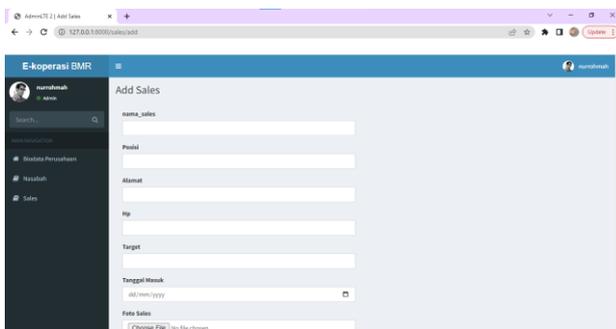
Tampilan detail sales merupakan halaman untuk melihat detail sales setelah user melakukan aksi klik tombol detail pada menu sales. Berikut ini merupakan antarmuka yang menampilkan detail sales, ditunjukkan pada gambar dibawah ini.



Gambar 7. Tampilan Detail Sales

### 6. Tampilan Add Sales

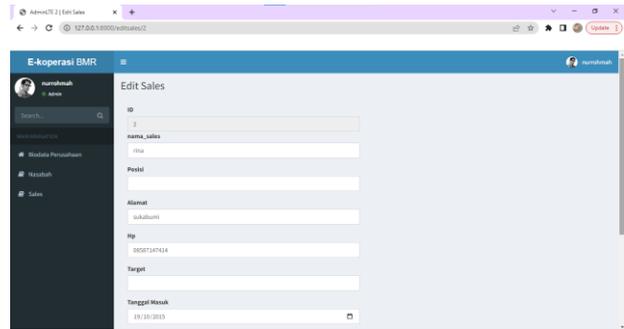
Halaman untuk menambah data sales setelah admin melakukan aksi klik tombol add pada menu sales.



Gambar 8. Tampilan Tambah Data Sales

### 7. Tampilan Edit Data Sales

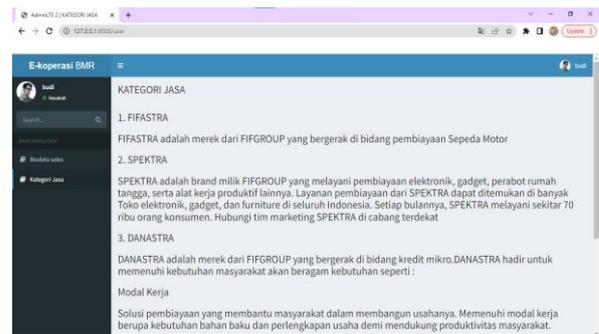
Tampilan halaman untuk admin mengedit data sales setelah melakukan aksi klik tombol edit pada menu sales.



Gambar 9. Tampilan Edit Data Sales

### 8. Tampilan Halaman Nasabah

Tampilan halaman awal nasabah setelah melakukan login.



Gambar 10. Tampilan Halaman Nasabah

### 9. Tampilan Detail Nasabah

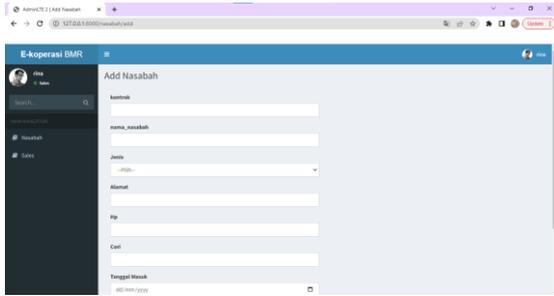
Halaman untuk melihat detail nasabah setelah user melakukan aksi klik tombol detail pada menu nasabah.



Gambar 11. Tampilan Detail Nasabah

### 10. Tampilan Tambah Data Nasabah

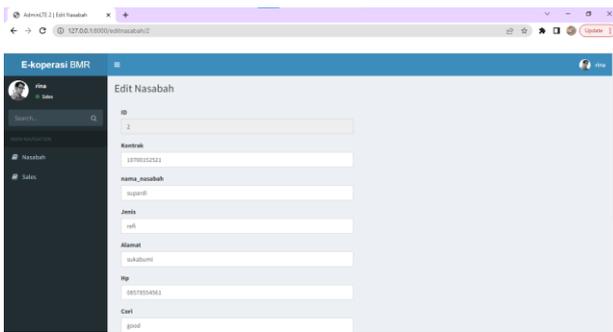
Halaman untuk menambah data nasabah setelah sales melakukan aksi klik tombol add pada menu nasabah.



Gambar 12. Tampilan Tambah Data Nasabah

11. Tampilan Edit Data Nasabah

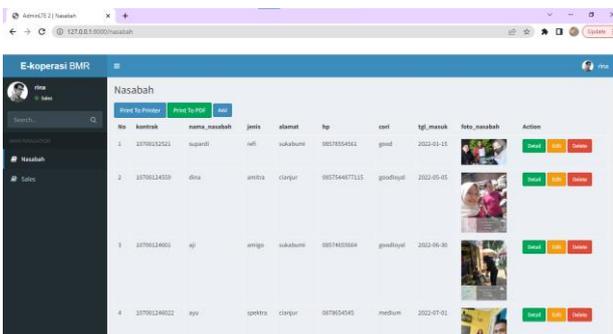
Tampilan halaman *untuk admin* mengedit data nasabah setelah melakukan aksi klik tombol edit pada menu nasabah.



Gambar 13. Tampilan Edit Data Nasabah

12. Tampilan Menu Nasabah

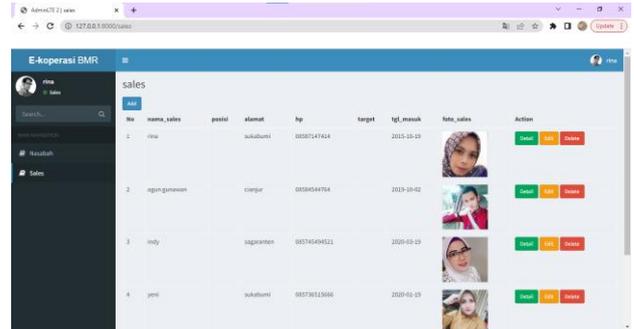
Menu nasabah merupakan menu yang menampilkan *form* nasabah.



Gambar 14. Tampilan Menu Nasabah

13. Tampilan Menu Sales

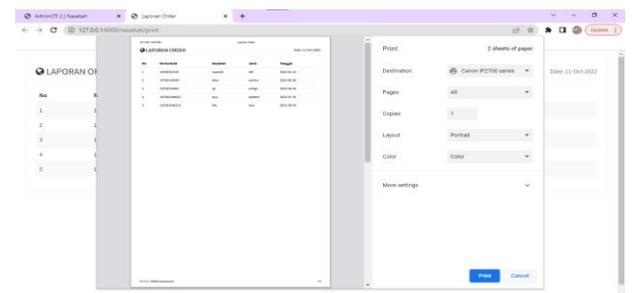
Menu sales merupakan menu yang menampilkan *form* sales.



Gambar 15. Tampilan Menu Sales

14. Tampilan Laporan Order

Menu laporan *order* merupakan menu yang menampilkan *form* laporan *order* bulanan.



Gambar 16. Tampilan Laporan Order

**Blackbox Testing**

Blackbox testing merupakan tahap pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tahu *ada* apa dibalik bungkus hitam nya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (interface), fungsionalitasnya. tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya.

Tabel 1. Hasil Pengujian *Black Box Register*

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu <i>register</i> pada <i>dashboard</i>	Menampilkan <i>form</i> <i>register</i> pengguna baru”	Sesuai Harapan	Valid
2	Mengosongkan salah satu <i>input</i> pada <i>form</i> <i>registrer</i>	Menampilkan pesan “bidang nama harus diisi”	Sesuai Harapan	Valid
3	Memasukan email yang sudah terdaftar	Menampilkan pesan”email sudah diambil”	Sesuai Harapan	Valid

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
4	Memasukan password hanya pada satu form	Menampilkan pesan "Konfirmasi kata sandi tidak cocok"	Sesuai Harapan	Valid
5	Mengetikan nama, email, password, dan konfirmasi password, kemudian klik tombol Register.	Sistem menerima akses register dan kemudian langsung login menampilkan halaman utama	Sesuai Harapan	Valid

Tabel 2. Hasil Pengujian Black Box Login

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Email dan Password yang dimasukan salah atau tidak terdaftar	Sistem akan menolak dan menampilkan pesan "tidak cocok dengan catatan kami"	Sesuai Harapan	Valid
2	Mengetikan email dan password yang terdaftar, kemudian klik tombol login	Sistem menerima akses login lalu akan menampilkan halaman utama	Sesuai Harapan	Valid

Tabel 3. Hasil Pengujian Black Box Fungsional Beranda

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu nasabah	Menampilkan tampilan menu nasabah	Sesuai harapan	Valid
2	Klik menu sales	Menampilkan tampilan menu sales	Sesuai harapan	Valid
3	Klik menu biodata perusahaan	Menampilkan tampilan halaman biodata perusahaan	Sesuai harapan	Valid
4	Klik biodata sales	Menampilkan daftar nama sales	Sesuai harapan	Valid

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
5	Klik kategori jasa	Menampilkan tampilan kategori jasa kredit yang ditawarkan	Sesuai Harapan	Valid

Tabel 4. Hasil Pengujian Black Box Menu Nasabah

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu nasabah	Menampilkan semua data nasabah yang telah diinput oleh sales	Sesuai harapan	Valid
2	Klik button add pada menu nasabah	Menampilkan form pengisian untuk menambah data nasabah	Sesuai harapan	Valid
3	Mengosongkan salah satu input pada form tambah nasabah	Sistem tidak akan mengambil data dan akan menampilkan kembali form tambah	Sesuai harapan	Valid
4	Mengosongkan input foto nasabah	Menampilkan pesan "kolom foto nasabah wajib diisi"	Sesuai harapan	Valid
5	Klik button detail pada saah satu data nasabah	Menampilkan detail data nasabah	Sesuai harapan	Valid
6	Klik Button edit pada menu nasabah	Menampilkan form edit data nasabah	Sesuai harapan	Valid
7	Klik button hapus pada salah satu data nasabah	Sistem akan menghapus data nasabah	Sesuai harapan	Valid
8	Klik button print to pdf pada menu nasabah	Otomatis akan mengunduh laporan order	Sesuai harapan	Valid
9	Klik button print to printer	Menampilkan halaman cetak laporan order	Sesuai harapan	Valid

Tabel 1. Hasil Pengujian *Black Box* Menu Sales

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu nasabah	Menampilkan semua data nasabah yang telah diinput oleh sales	Sesuai harapan	Valid
2	Klik button <i>add</i> pada menu sales	Menampilkan <i>form</i> pengisian untuk menambah data sales	Sesuai harapan	Valid
3	Mengosonkan salah satu <i>input</i> pada <i>form</i> tambah sales	Sistem tidak akan mengambil data dan akan menampilkan kembali <i>form</i> tambah	Sesuai harapan	Valid
4	Mengosonkan <i>input</i> foto sales	Menampilkan pesan "kolom foto sales wajib diisi"	Sesuai harapan	Valid
5	Klik button detail pada saah satu data sales	Menampilkan detail data sales	Sesuai harapan	Valid
6	Klik Button edit pada menu sales	Menampilkan <i>form</i> edit data sales	Sesuai harapan	Valid
7	Klik button hapus pada salah satu data sales	Sistem akan menghapus data sales	Sesuai harapan	Valid

Tabel 2. Hasil Pengujian *Black Box* Biodata Perusahaan

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu biodata perusahaan	Menampilkan biodata perusahaan	Sesuai harapan	Valid

Tabel 3. Hasil Pengujian *Black Box* Kategori Jasa

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu biodata perusahaan	Menampilkan biodata perusahaan	Sesuai harapan	Valid

Tabel 4. Hasil Pengujian *Black Box* Biodata Sales

No	Skenario Pengujian	Hasil Diharapkan	Hasil Uji	Simpulan
1	Klik menu biodata Sales	Menampilkan biodata sales	Sesuai harapan	Valid

#### IV. KESIMPULAN

Dengan adanya Perancangan Aplikasi E-koperasi Berbasis *Web* pada PT Bersama Makmur Raharja ini diupayakan dapat menyelesaikan berbagai permasalahan yang *di* hadapi oleh para pekerja *di* perusahaan tersebut dengan adanya sistem baru yang disusun oleh penulis ini. Beberapa kesimpulan yang dapat diambil dari pembuatan Sistem Informasi Monitoring Sales ini antara lain:

1. Dengan adanya sistem informasi ini dapat mempercepat distribusi data dari *admin* ke sales sehingga laporan pengiriman penjualan lebih efektif kepada mitra perusahaan.
2. Dengan adanya sistem informasi ini data dapat diakses secara terkini oleh siapapun dan dimanapun tanpa terkendala *admin* yang tidak hadir.

#### REFERENSI

- [ 1 ] Kurniawan, A. (2019, Agustus 01). Analisis – Pengertian, Contoh, Tahap, Tujuan, Para Ahli. Retrieved from gurupendidikan.co.id: <https://www.gurupendidikan.co.id/analisis/>
- [ 2 ] Wirid Winduro, 2018. PENGERTIAN MONITORING DAN EVALUASI available at [https://www.academia.edu/27658022/PENGERTIAN\\_MONITORING\\_DAN\\_EVALUASI](https://www.academia.edu/27658022/PENGERTIAN_MONITORING_DAN_EVALUASI) [Diakses 10 April 2019].
- [ 3 ] Rocketmanajemen. (2019, July 08). Pengertian *MySQL*. Retrieved from rocketmanajemen.com: <https://rocketmanajemen.com/definisi-mysql/>
- [ 4 ] Tahir, T. B., Rais, M., & Sirad, M. A. H. (2019). Aplikasi Point *OF* Sales Menggunakan *Framework Laravel*. JIKO (Jurnal Informatika dan Komputer), 2(2), 55-59.
- [ 5 ] Fahriya, K. (2018). Rancang Bangun Simawa (Sistem Informasi Rusunawa) Berbasis *Web Application* Menggunakan *Framework Laravel*. Jurnal Manajemen Informatika, 8(2).
- [ 6 ] Fridayanthie, E. W., & Mahdiati, T. (2016). Rancang bangun sistem informasi permintaan atk berbasis intranet (*studi kasus: kejaksanaan negeri rangkasbitung*). Jurnal khatulistiwa informatika, 4(2).
- [ 7 ] Mustamiin, M., Ismantohadi, E., Ghozali, A. L., Darsih, D., & Inara, L. N. (2020). Rancang Bangun Sistem Manajemen Soal Dan Ujian

- Berbasis *Website* Menggunakan *Framework Laravel*. *IKRA-ITH INFORMATIKA: Jurnal Komputer dan Informatika*, 4(1), 58-63.
- [ 8 ] Soepeno, B. (2014). Penggunaan Aplikasi CMS *Wordpress Untuk Merancang Website* Sebagai Media Promosi pada Maroon Wedding Malang. *Jurnal Akuntansi, Ekonomi dan Manajemen Bisnis*, 2 (1), 63-69.
- [ 9 ] “Pengertian *Query*. *Query* adalah *perintah-perintah untuk mengakses data pada sistem basis data - PDF Free Download*,” Docplayer.info, 2015. <https://docplayer.info/69602940-Pengertian-query-query-adalahperintahperintah-untuk-mengakses-data-pada-sistem-basis-data.html> (accessed Sep. 21, 2021).
- [ 10 ] Pakpahan, A. F. (2020). *Dasar-Dasar Pengembangan Aplikasi Web Modern dengan Framework Laravel*. Media Sains Indonesia.