

Perancangan Database Point of Sales Apotek Dengan Menerapkan Model Data Relasional

Rusydi Umar¹, Abdul Hadi², Panggah Widiandana³, Fahmi Anwar⁴
Muhammad Jundullah⁵, Ahmad Ikrom⁶

¹ Program Studi Teknik Informatika, Universitas Ahmad Dahlan
^{2,3,4,5,6} Program Studi Magister Teknik Informatika, Universitas Ahmad Dahlan
Jl. Prof. Dr. Soepomo, S.H, Warungboto, Yogyakarta 55164
¹rusydi_umar@rocketmail.com, ²abdu1808048032@webmail.uad.ac.id
³panggah1808048029@webmail.uad.ac.id

Abstract

Database design is a process of designing data needed to support various system designs. The benefit of a database in Point of Sales drug sales in the Jogja Pharmacy enables storing, changing and returning all data quickly and easily. Problems in database design are different viewpoints from the designer, programmer and end-user. Because of that, the design is needed a methodology the process of designing a good database starting from procedures, techniques, tools and documentation. The database planning method used in Research and Development (R & D) and the database design method uses the Database Life Cycle (DBLC) variable research is POS pharmacy. Research aspects include requirement analysis, conceptual design, logical design and physical design. The final result of this research is to create 8 types of conceptual consists of drugs, drugs, customers, stores, users, detailed details, and temporary details.

Keywords: database, apotek, POS, DBLC, relational data

1. PENDAHULUAN

Teknologi sangat memberikan manfaat dan membantu kegiatan manusia sehari-hari. Saat ini teknologi dapat membantu manusia dari permasalahan banyaknya data yang tersimpan pada sistem, database pada teknologi informasi dapat menata dengan rapi data di dalam penyimpanan. *Database* merupakan penggabungan beberapa tabel ter-manage sehingga dapat dipakai beberapa pemakai atau program lain untuk diakses dan dimanipulasi sesuai kebutuhan. Metadata pada *database* memberikan info yang tertata dan dapat dikelola dengan uraian kepemilikan data yang lengkap[1]. Proses pembuatan desain untuk mendukung operasional dan tujuan organisasi disebut perancangan *database*[2]. *Database* dapat dimanfaatkan untuk menyimpan dan merubah sehingga data tersebut dapat dipanggil kembali dengan mudah.

Perkembangan sistem *database* beriring dengan kemajuan sistem informasi. Siklus perkembangan *database* tidak harus berurutan tetapi juga melibatkan tahapan evaluasi dari sisi pengguna sistem informasi. Permasalahan dalam perancangan *database* adalah sudut pandang dari sisi perancang, *programmer* dan *enduser* cenderung berbeda pemahaman. Metodologi perancangan merupakan struktur yang menggunakan prosedur, teknik, peralatan dan dokumentasi. Fase perancangan mengarahkan desainer untuk menggunakan teknik sesuai dan efektif dalam proyek perancangan, sehingga membantu desainer dalam perencanaan, pengelolaan, pengaturan dan evaluasi pengembangan *database* [3].

Perancangan *database* terdiri dari 4 proses yaitu analisis kebutuhan, desain konseptual, desain logikal dan fisik. Analisis kebutuhan adalah memahami kebutuhan dari sistem informasi yang mau dibuat seperti *system requirement*, *rule business*, kebutuhan fungsional dan kebutuhan nonfungsional. Desain konseptual adalah proses perancangan model data terbebas dari semua pertimbangan fisik. Desain logikal adalah perancangan model data berdasarkan data spesifik terbebas dari DBMS sedangkan fisik hasil dari uraian implementasi *database* yang disimpan dan saling berhubungan. Relasi adalah representasi semua *database* dibuat dalam bentuk tabel dan kolom secara dua dimensi[4].

Penelitian terdahulu merupakan bentuk manifestasi dari kemajuan teknologi dengan menghasilkan manfaat besar bagi masyarakat. Penelitian yang telah dibuat oleh Mohammad

Jamiluddin (2019) dari Fakultas Teknologi Informasi, Universitas KH. A. Wahab Hasbullah dengan judul perancangan POS yang terintegrasi dengan pergudangan. Metode perancangan sistem menggunakan metode SWOT (*Strengths, Weaknesses, Opportunities, Threats*). Pengembangan aplikasi menggunakan Bahasa pemodelan UML, implementasi menggunakan PHP untuk server dan *basic4android* untuk client. Hasil penelitian berupa sistem informasi *Point of sales* (POS) untuk memberikan laporan yang valid, baik laporan transaksi maupun laporan barang[5]. Penelitian selanjutnya oleh Arif Cahyo Baskoro (2018) dari Program Studi Informatika, Universitas Muhammadiyah Surakarta dengan judul pembuatan aplikasi penjualan alat teknik pada CV Arba menggunakan platform WEB. Metode yang digunakan adalah *Waterfall*, pemrograman menggunakan PHP dan MYSQL. Hasil penelitian 82,5% layak untuk digunakan untuk mempermudah pengelolaan transaksi penjualan secara efektif dan efisien[6]. Penelitian yang telah dibuat Hendy Thomas Herman (2016) dari Program Studi Informatika, Universitas Kristen Petra, Surabaya, penelitian berjudul sistem informasi POS pada rumah makan Rinjani. Aplikasi yang dibuat berbasis web yang dapat diakses secara mobile menggunakan jaringan lokal, menggunakan framework PHP, Javascript, HTML dan mySQL. Penelitian menghasilkan sistem informasi yang menampilkan menu pemesanan, pengaturan user, member dan reservasi, menambah promosi dan menampilkan laporan penjualan dan stock[7]. Penelitian yang telah dibuat oleh Nicko Happy Atmaja (2015) judul perancangan aplikasi apotek, studi kasus: Apotek Ashara Husada Sidoarjo. Metode menggunakan model *waterfall* berbasis desktop, hasil penelitian aplikasi yang dibuat mampu mengurangi jumlah penolakan penjualan obat, mampu memberikan informasi pembelian obat, penjualan obat dan stock obat, aplikasi mampu menampilkan grafik transaksi penjualan, pembelian, stok, dan penolakan obat[8].

2. METODE PENELITIAN

Metode yang dipakai untuk membangun database menggunakan R&D. Menurut Sugiyono (2009) R&D merupakan sistematika alur *research* untuk menghasilkan *output* dan pengujian untuk membuktikan efektifnya sebuah produk[9]. Produk tidak sebatas benda fisik seperti hardware melainkan berbentuk perangkat lunak (*software*). Perancangan *database* menggunakan *Database Life Cycle* (DBLC) dengan subjek penelitian adalah POS Apotek dengan model data relasional. Aspek penelitian meliputi analisis kebutuhan, rancangan konseptual, rancangan logikal, dan rancangan fisik sesuai dengan Gambar 1.



Gambar 1. Rancangan *database*

3. HASIL DAN PEMBAHASAN

Perancangan *database* sangat penting dalam pembuatan sistem informasi karena dengan adanya perancangan dapat memenuhi kebutuhan saat ini dan masa yang akan datang. Perancangan konseptual akan menunjukkan *entity* dan *relasi* berdasarkan proses yang diinginkan oleh user. Penentuan *entity* dan relasi dengan mengambil data dan *role business* untuk dijadikan bahan analisis menyesuaikan kebutuhan sistem informasi yang dibuat. Tahapan yang digunakan untuk perancangan *database* dibagi dalam 4 tahap yaitu analisis kebutuhan, perancangan *database* konseptual, perancangan *database* logikal dan perancangan *database* fisik.

3.1. Analisis Kebutuhan

Analisis kebutuhan *user point of sale* di salah satu apotek diantaranya:

- Sistem dapat menampilkan halaman login untuk kasir.
- Sistem dapat menampilkan halaman dashboard ketika berhasil login.
- Sistem dapat menginput obat yang diminta oleh pelanggan atau konsumen.

- d. Sistem dapat menginput harga obat yang telah dibeli oleh kasir.
- e. Sistem dapat menambah inputan obat sebanyak obat yang diperlukan oleh konsumen atau pelanggan.
- f. Sistem dapat menjumlah total harga dari beberapa item obat yang telah dibeli oleh pelanggan atau konsumen.
- g. Sistem dapat mencetak nota sebagai bukti transaksi penjualan obat pada apotek tersebut.
- h. Sistem dapat memberikan potongan harga bagi pelanggan yang sudah mendaftarkan sebagai member.

3.2 Desain Konseptual

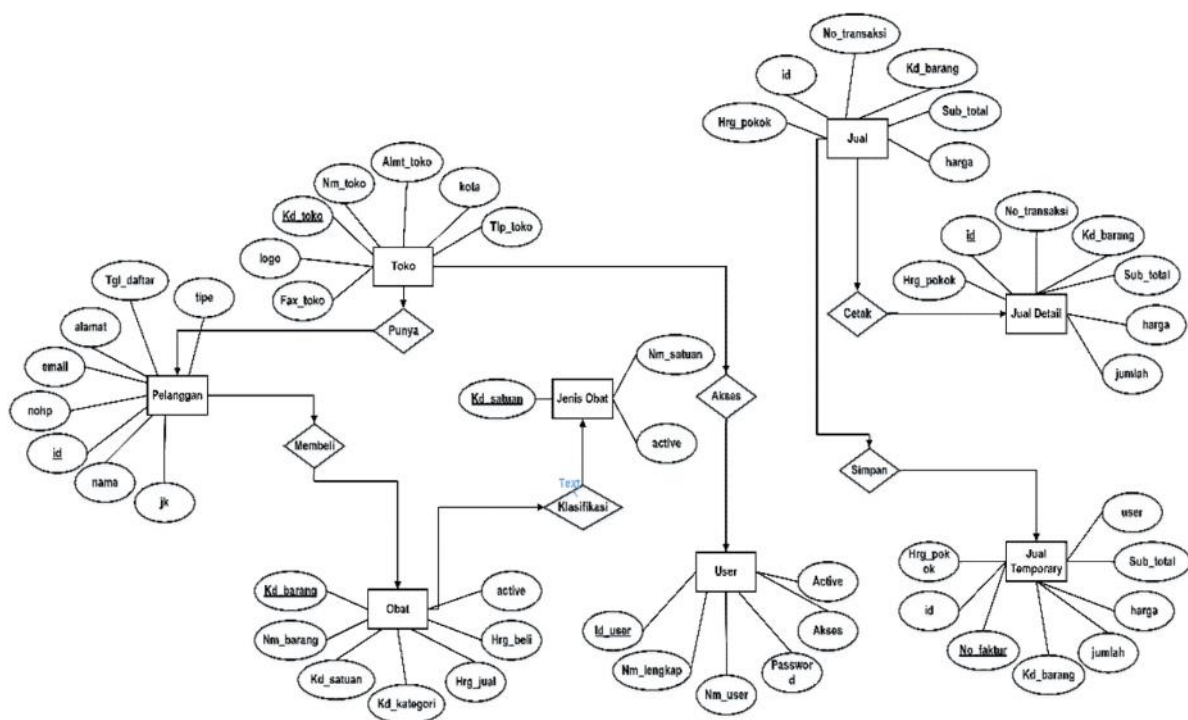
Desain konseptual *database*, yaitu proses pembentukan model yang berasal dari informasi yang didapatkan dalam proses analisis kebutuhan dari keseluruhan aspek data yang diperlukan untuk perancangan sistem *database* dan digunakan sebagai sumber informasi untuk tahapan desain logikal sesuai dengan Tabel 1. Identifikasi tipe entitas.

Tabel 1. Identifikasi Tipe Entitas

| No. | Nama Entitas | Deskripsi | Kejadian |
|-----|---------------------|---|--|
| 1 | Jenis Obat | Mendeskripsikan nama satuan pada obat yang telah diinputkan | Setiap konsumen dapat membeli obat sesuai dengan jenis obat yang tersedia pada apotek |
| 2 | Nama Obat | Mendeskripsikan nama obat yang ada dalam aplikasi POS apotek | Setiap konsumen dapat membeli obat sesuai dengan nama obat yang tersedia pada apotek |
| 3 | Nama Toko | Mendeskripsikan data toko yang menggunakan aplikasi POS apotek | Setiap apotek dapat menggunakan POS apotek |
| 4 | Pelanggan | Mendeskripsikan pelanggan atau member pada apotek tersebut | Setiap pelanggan atau member akan diberikan potongan harga |
| 5 | Nama User | Mendeskripsikan data karyawan pada sistem setiap tindakan | Setiap karyawan login untuk sistem keamanan dan kerahasiaan POS apotek |
| 6 | Penjualan | Mendeskripsikan penjualan obat ke konsumen | Setiap penjualan dibuat penyimpanan agar dapat dilaporkan |
| 7 | Penjualan Detail | Mendeskripsikan laporan pembelian obat kepada pemasok secara detail | Setiap penjualan akan diberikan laporan berupa faktur pembelian |
| 8 | Penjualan Temporary | Mendeskripsikan tempat sementara untuk menampung transaksi | Setiap transaksi disimpan pada sebuah <i>temporary</i> yang akan menyimpan sementara transaksi |

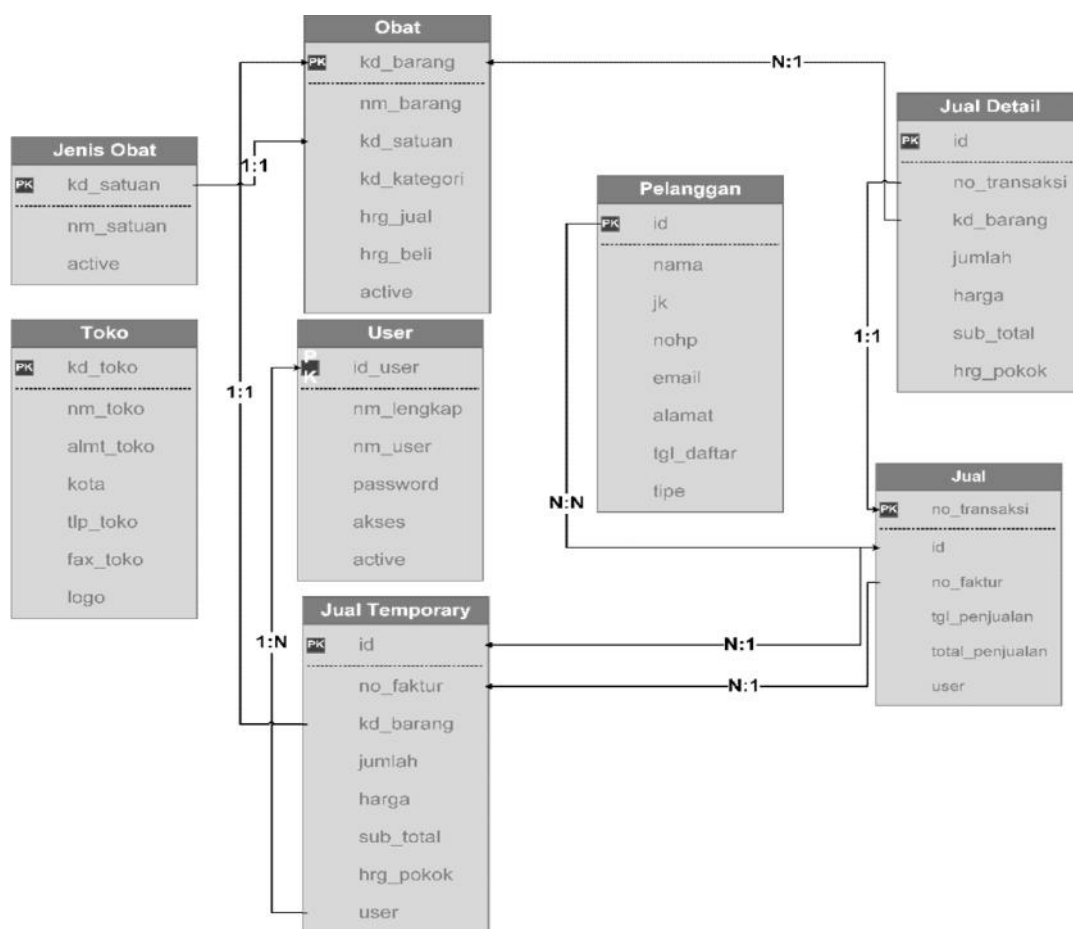
3.3 Desain Logikal

Desain logikal yaitu proses pembuatan model dari informasi yang digunakan perusahaan berdasarkan model dan data spesifik. Deskripsi implementasi *database* berdasarkan hasil desain logikal dengan *Entity Relationship Diagram* (ERD) pada *Database Management System* (DBMS) sesuai pada Gambar 2.



Gambar 2. Diagram relasi entitas

Perancangan database relasional yang baik adalah menghapus yang tidak sesuai sebagian dari model data konseptual. Berikut pembuatan *mapping table* sesuai Gambar 3.



Gambar 3. Mapping table

3.4 Desain Fisikal

Desain fisikal merupakan tahapan implementasi dari pembuatan database logis yang tersimpan secara fisik pada harddisk sesuai dengan DBMS yang digunakan. Salah satu dari DBMS adalah *My Structured Query Language* (MySQL), berikut Tabel 2 sampai Tabel 9, hasil pembuatan 8 entitas apotek.

Tabel 2. Entitas Jenis Obat

| Atribut | Deskripsi | Tipe Data dan Panjang |
|-----------|-------------------|-----------------------|
| kd_satuan | ID Jenis Obat | Varchar [6] |
| nm_satuan | Nama Jenis Obat | Varchar [25] |
| active | Status Jenis Obat | Char [1] |

Tabel 3. Entitas Obat

| Atribut | Deskripsi | Tipe Data dan Panjang |
|-------------|------------------|-----------------------|
| kd_barang | ID Obat | Varchar [5] |
| nm_barang | Nama Obat | Varchar [50] |
| kd_satuan | ID Jenis Obat | Varchar [10] |
| kd_kategori | Kategori Obat | Varchar [10] |
| hrj_jual | Harga Jual Obat | Int [11] |
| hrj_beli | Harga Beli Obat | Int [11] |
| active | Status Nama Obat | Char [1] |

Tabel 4. Entitas Pelanggan

| Atribut | Deskripsi | Tipe Data dan Panjang |
|------------|-------------------------------|-----------------------------|
| id | ID Pelanggan | Double |
| nama | Nama Pelanggan | Varchar [35] |
| jk | Jenis Kelamin Pelanggan | Enum [Laki-laki, Perempuan] |
| nohp | Nomor Handphone Pelanggan | Varchar [15] |
| email | Email Pelanggan | Varchar [35] |
| alamat | Alamat Pelanggan | Text |
| tgl_daftar | Tanggal Pendaftaran Pelanggan | Date |
| tipe | Tipe Pelanggan | Char [2] |

Tabel 5. Entitas Toko

| Atribut | Deskripsi | Tipe Data dan Panjang |
|-----------|--------------------|-----------------------|
| kd_toko | ID Toko | Varchar [15] |
| nm_toko | Nama Toko | Varchar [30] |
| almt_toko | Alamat Toko | Varchar [150] |
| kota | Kota Toko | Varchar [30] |
| tlp_toko | Nomor Telepon Toko | Varchar [15] |
| fax_toko | Fax Toko | Varchar [15] |
| logo | Logo Toko | Varchar [50] |

Tabel 6. Entitas User

| Atribut | Deskripsi | Tipe Data dan Panjang |
|------------|----------------|-----------------------|
| id_user | ID User | Varchar [6] |
| nm_lengkap | Nama User | Varchar [30] |
| nm_user | Nama Username | Varchar [25] |
| password | Password User | Varchar [35] |
| akses | Hak akses User | Varchar [15] |
| active | Status User | Char [1] |

Tabel 7. Entitas Jual

| Atribut | Deskripsi | Tipe Data dan Panjang |
|-----------------|------------------------|-----------------------|
| id | ID Pelanggan | Double |
| no_transaksi | Nomor Penjualan | Double |
| no_faktur | Nomor Faktur Penjualan | Varchar [15] |
| tgl_penjualan | Tanggal Penjualan | Date |
| total_penjualan | Total Penjualan | Int [11] |
| user | Penginput | Varchar [15] |

Tabel 8. Entitas Jual Detail

| Atribut | Deskripsi | Tipe Data dan Panjang |
|--------------|-----------------------|-----------------------|
| id | ID Pelanggan | Double |
| no_transaksi | Nomor Penjualan | Double |
| kd_barang | Kode Barang | Varchar [15] |
| jumlah | Jumlah Penjualan | Int [11] |
| harga | Harga Penjualan | Int [11] |
| sub_total | Sub Penjualan | Int [11] |
| hrg_pokok | Harga Pokok Penjualan | Int [11] |

Tabel 9. Entitas Jual Temporary

| Atribut | Deskripsi | Tipe Data dan Panjang |
|-----------|-----------------------|-----------------------|
| id | ID Pelanggan | Double |
| no_faktur | Nomor Faktur | Double |
| kd_barang | Kode Barang | Varchar [15] |
| jumlah | Jumlah Penjualan | Int [11] |
| harga | Harga Penjualan | Int [11] |
| sub_total | Sub Penjualan | Int [11] |
| hrg_pokok | Harga Pokok Penjualan | Int [11] |
| user | User yang menginput | Varchar [15] |

Data definition Language (DDL) adalah kumpulan beberapa perintah SQL untuk memvisualkan desain database secara menyeluruh. Beberapa fungsi perintah SQL yaitu *create* (membuat), *alter* (merubah) dan *drop* (menghapus), berikut *script* DDL yang diimplementasikan pada apotek.

a. SQL Create Table Jenis Obat

```
CREATE TABLE 'jenis_obat' (
  'kd_satuan' varchar(6) NOT NULL,
  'nm_satuan' varchar(25) NOT NULL,
  'active' char(1) NOT NULL
) ENGINE = InnoDB DEFAULT CHARSET=latin1;
```

b. SQL Create Table Obat

```
CREATE TABLE 'obat' (
  'kd_barang' varchar(5) NOT NULL,
  'nm_barang' varchar(50) NOT NULL,
  'kd_satuan' varchar(10) NOT NULL,
  'kd_kategori' varchar(10) NOT NULL,
  'hrng_jual' int(11) NOT NULL,
  'hrng_beli' int(11) NOT NULL,
  'active' char(1) NOT NULL
) ENGINE = InnoDB DEFAULT CHARSET=latin1;
```

c. SQL Create Table Pelanggan

```
CREATE TABLE `pelanggan` (
  `tgl_daftar` date NOT NULL,
  `tipe` char(2) NOT NULL,
  `ID` double NOT NULL,
  `Nama` varchar(35) NOT NULL,
  `Jk` enum('Laki-Laki','Perempuan') NOT NULL,
  `NoHp` varchar(15) NOT NULL,
  `Email` varchar(35) NOT NULL,
  `Alamat` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

d. SQL Create Table Toko

```
CREATE TABLE `nama_toko` (
  `kd_toko` varchar(15) NOT NULL,
  `nm_toko` varchar(30) NOT NULL,
  `almt_toko` varchar(150) NOT NULL,
  `kota` varchar(30) NOT NULL,
  `tlp_toko` varchar(15) NOT NULL,
  `fax_toko` varchar(15) NOT NULL,
  `logo` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

e. SQL Create Table User

```
CREATE TABLE `nama_user` (
  `id_user` varchar(6) NOT NULL,
  `nm_lengkap` varchar(30) NOT NULL,
  `nm_user` varchar(25) NOT NULL,
  `password` varchar(35) NOT NULL,
  `akses` varchar(15) NOT NULL,
  `active` char(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

f. SQL Create Table Jual

```
CREATE TABLE `penjualan` (
  `id` double NOT NULL,
  `no_transaksi` double NOT NULL,
  `no_faktur` varchar(15) NOT NULL,
  `tgl_penjualan` date NOT NULL,
  `total_penjualan` int(11) NOT NULL,
  `user` varchar(15) NOT NULL,
  `tipe` char(2) NOT NULL,
  `jenis` enum('debit','kredit','','') NOT NULL,
  `id_pelanggan` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

g. SQL Create Table Jual Detail

```
CREATE TABLE `penjualan_detail` (
  `id` double NOT NULL,
  `no_transaksi` double NOT NULL,
  `kode_obat` varchar(15) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `harga` int(11) NOT NULL,
  `sub_total` int(11) NOT NULL,
  `hrng_pokok` int(11) NOT NULL,
  `tipe` char(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

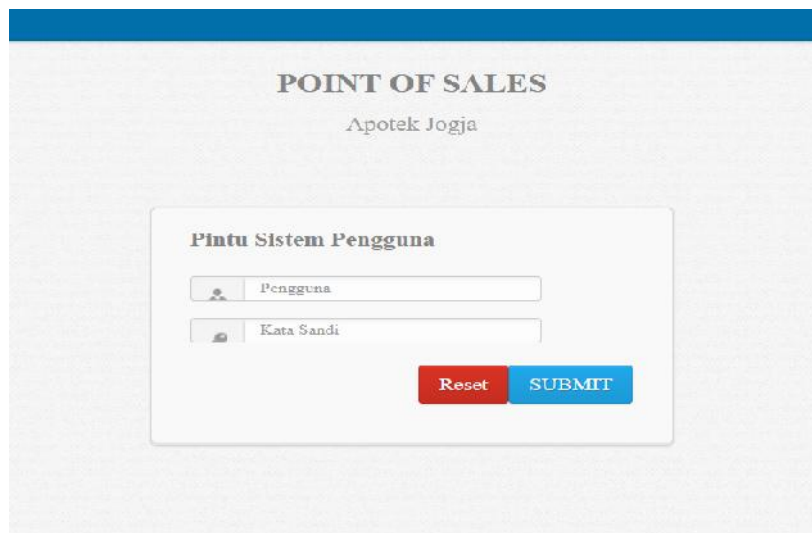
h. SQL Create Table Jual Temporary

```
CREATE TABLE `penjualan_tmp` (
  `id` double NOT NULL,
  `no_faktur` double NOT NULL,
  `kode_obat` varchar(15) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `harga` int(11) NOT NULL,
  `sub_total` int(11) NOT NULL,
  `hrng_pokok` int(11) NOT NULL,
  `user` varchar(15) NOT NULL
)
```

3.5 Implementasi Input dan Output Database dalam Aplikasi

a. Halaman Login

Halaman login diperlukan untuk keamanan aplikasi, login ini akan tampil pertama kali pada sebuah aplikasi, sehingga orang yang akan melakukan transaksi tidak dapat sembarangan orang melakukan transaksi, POS ini ada beberapa level yaitu admin, kasir dan kepala keuangan. Hasil halaman *login* seperti pada Gambar 4.

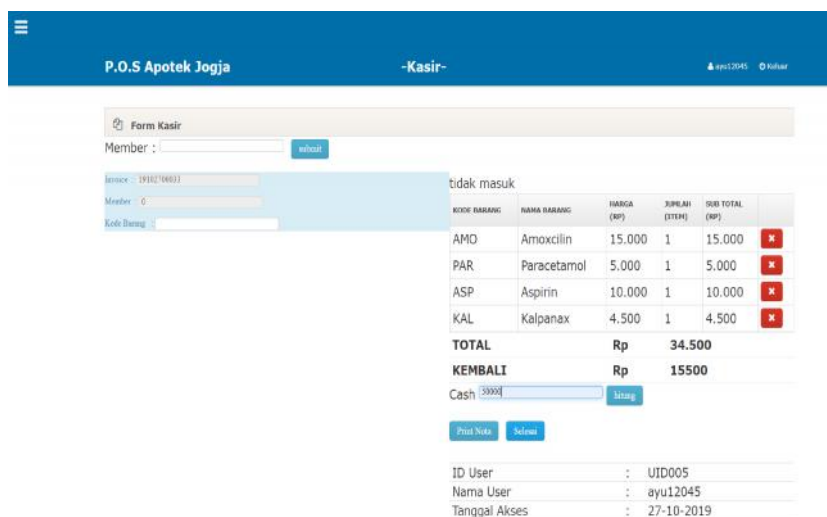


Gambar 4. Halaman *Login* POS ApJog

Gambar 4 memiliki dua inputan dan dua *button* yaitu input *username*, *password* dan tombol *reset* jika ingin membatalkan pengisian *username* dan *password* serta tombol *submit* untuk login ke dalam sebuah sistem POS ApJog.

b. Halaman Penjualan

Halaman penjualan digunakan sebagai tempat proses transaksi penjualan obat dalam halaman ini akan menampilkan nama barang yang ingin dibeli dan harga. Tampilan halaman penjualan dapat dilihat pada Gambar 5.



Gambar 5. Halaman penjualan

Gambar 5 merupakan halaman penjualan yang terdapat *form* input yang digunakan untuk memasukkan obat yang akan dibeli dan jumlah uang yang telah dibayar oleh pasien. Tombol print juga digunakan apabila kasir akan mencetak nota.

4. KESIMPULAN

Berdasarkan penelitian rancangan *database* dengan menggunakan metode DBLC didapatkan hasil sebagai berikut:

1. Analisis Kebutuhan
Rule business dan kebutuhan user.
2. Desain Konseptual
Entitas yang diperlukan sebanyak 8 entitas, berisikan keterangan dan kegiatan entitas.
3. Desain Logikal
Menghasilkan relasi model dan logikal untuk mempresentasikan entitas, relasi dan *attribute* yang telah diidentifikasi sebelumnya.
4. Desain Fisikal
Perancangan *database* menggunakan MySQL, jumlah tabel dalam *database* APJOG ada 8 yaitu jenis obat, obat, pelanggan, toko, user, jual, jual detail dan jual *temporary*.
5. *Database* dengan model relasional dalam bentuk dua dimensi yang saling berkaitan.
6. Perancangan basis data dapat diterapkan pada aplikasi ApJog seperti Login yang terintegrasi dengan tabel user pada *database*.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Direktur dan Kaprodi Program Pascasarjana Teknik Informatika Universitas Ahmad Dahlan yang (UAD) yang telah memberikan dukungan pada penelitian ini. Terimakasih kepada dosen dan staff di MTI UAD dan *reviewer* semoga penelitian ini dapat disajikan melalui publikasi jurnal sesuai harapan publik.

BAHAN REFERENSI

- [1] Fathansyah, 1999, *Basis Data*. Informatika, Bandung.
- [2] Whitehorn, M. Dan Marklyn, B., 2003, *Seluk Beluk Database Relasional*, Edisi ke-2. Erlangga, Jakarta.
- [3] Mannino, M. V., 2001, *Database Application Development and Design*. McGraw- Hill, New York.
- [4] Laudon, K. C., Laudon, J. P., 2005, *Management Information Systems: Managing the Digital Firm, 8th Edition*, Prentice Hall, New Jersey
- [5] Kasus, S., Maan, P. T., & Shiddiq, G., 2019. *Rancang Bangun Point of sales Terintegrasi Dengan Sistem Gudang*. (1), 57–67.
- [6] Baskoro, A. C., Informatika, P. S., Komunikasi, F., Informatika, D. A. N., & Surakarta, U. M. (2018). *Sistem informasi penjualan alat teknik berbasis web pada cv arba kurnia teknik*.
- [7] Herman, H. T., Rostianingsih, S., & Setiawan, A, 2016. *Pembuatan Aplikasi Point of sales untuk Rumah Makan Dapur Rinjani*. *Infra*, 4, 6.
- [8] Happy, N., Dewiyani, A., & Ayuningtyas, S., 2015. *RANCANG BANGUN APLIKASI APOTEK (Studi Kasus : Apotek Ashara Husada Sidoarjo)*. *Jsika*, 4(2), 1–9.
- [9] Sugiyono., 2009, *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R & D*, Alfabeta, Jakarta
- [10] Nasution, Muhammad Irwan Padli, 2014, Keunggulan Kompetitif dengan Teknologi Informasi. *Jurnal Elektronik*