

Rancang Bangun Sistem *Electronic Prescribing* Dokter dengan Menggunakan *Codeigniter*

Dian Rakasiwi¹, Ria Arafiah², Fariani Hermin Indiyah³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta, Rawamangun, Jakarta Timur, 13220

dian_rakasiwi98@yahoo.co.id¹, fherminin@fmipa.unj.ac.id², riarafiah@fmipa.unj.ac.id³

Abstrak—Dalam perkembangan dunia kesehatan di Indonesia, sistem pembuatan resep obat masih banyak mengalami masalah seperti kesalahan dalam mengartikan resep obat yang ditulis tangan oleh dokter dan masih banyak dijumpai antrean yang panjang dalam pengambilan obat di setiap rumah sakit. Maka dari itu, penulis akan merancang dan membangun sistem *e-prescribing* (*electronic prescribing*) atau peresepan elektronik dengan menggunakan *framework codeigniter* dalam pembuatan resep obat agar waktu penebusan obat lebih efektif dan efisien. Cara kerja sistem ini adalah dokter akan *menginput* obat-obatan yang dibutuhkan untuk memulihkan kondisi pasien pada halaman web, kemudian resep yang telah *diinput* oleh dokter akan terintegrasi dengan apoteker sehingga apoteker dapat langsung mengetahui obat apa saja yang harus diracik tanpa perlu menunggu pasien yang menyerahkan resep dokter terlebih dahulu. Sistem *electronic prescribing* menggunakan *framework codeigniter* karena performanya yang cepat dan menggunakan *pattern MVC* (*Model View Controller*), sehingga struktur *code* menjadi lebih terstruktur dan memiliki standart. Pengembangan sistem *electronic prescribing* menggunakan teori pengembangan perangkat lunak model *waterfall* dengan tahap yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, serta penerapan program dan pemeliharaan.

Keywords— Resep; *electronic prescribing*; *codeigniter*; model *waterfall*

I. LATAR BELAKANG

Resep merupakan permintaan tertulis dari seorang dokter kepada apoteker untuk membuat dan menyerahkan obat kepada pasien. Resep menjadi bagian yang sangat penting dalam proses pemulihan pasien. Hal ini dikarenakan apabila terjadi kesalahan pada penulisan resep maka akan menimbulkan dampak yang fatal terhadap kondisi pasien. Di rumah sakit yang masih menggunakan sistem peresepan konvensional, mekanisme berobat dimulai dengan seorang pasien yang datang ke rumah sakit. Kemudian pasien mendaftar di loket pendaftaran. Petugas akan memberikan nomer urut kepada pasien dan mengirim rekam medis milik pasien ke dokter yang dituju.

Rekam medis membantu dokter untuk melihat riwayat penyakit dan obat yang pernah diberikan sehingga apabila pasien menderita penyakit yang sama maka dokter cukup menulis resep dan dalam dosis yang sama agar pasien tidak mengalami overdosis. Dokter akan memberikan resep kepada pasien, setelah itu pasien akan menyerahkan resep tersebut ke

apotek dan menunggu giliran dipanggil untuk mengambil obat miliknya.

Dalam perkembangan dunia kesehatan di Indonesia, sistem pembuatan resep obat masih banyak mengalami masalah seperti kesalahan dalam mengartikan resep obat yang ditulis tangan oleh dokter dan masih banyak di jumpai antrean yang panjang dalam pengambilan obat di setiap rumah sakit. Hal ini membuat waktu pengambilan obat tidak efisien. Banyak pasien yang tertidur saat menunggu giliran namanya dipanggil, karena pasien sedang sakit dan harus menunggu antrean panjang dalam pengambilan obat. Ada juga beberapa kasus kesalahan dalam penentuan dosis obat atau pemberian obat yang salah. Ika Susanti mengatakan bahwa penelitian di Amerika yang memperhitungkan kematian akibat kesalahan obat, kebanyakan terjadi pada saat fase *prescribing* atau peresepan yang diakibatkan dari kurangnya dalam pengetahuan, komunikasi yang buruk, dan kurangnya mempertimbangkan informasi penting pasien [6]. Fase *prescribing* merupakan fase penulisan resep oleh dokter.

Perancangan dan pengimplementasian sistem resep obat sudah pernah dilakukan oleh Puspa Setia Pratiwi dan Andri Lestari namun belum menggunakan sistem penomoran dalam pengambilan obat secara otomatis sehingga pasien harus ke apotek terlebih dahulu untuk mengambil nomer pengambilan obat kemudian menunggu antrean pengambilan obat. Maka dari itu, penulis akan merancang dan membangun sistem *e-prescribing* (*electronic prescribing*) atau peresepan elektronik dengan menggunakan *framework codeigniter* dalam pembuatan resep obat agar waktu penebusan obat lebih efektif dan efisien. Cara kerja sistem ini adalah dokter akan *menginput* obat-obatan yang dibutuhkan untuk memulihkan kondisi pasien pada halaman web, kemudian resep yang telah *diinput* oleh dokter akan terintegrasi dengan apoteker sehingga apoteker dapat langsung mengetahui obat apa saja yang harus diracik tanpa perlu menunggu pasien yang menyerahkan resep dokter terlebih dahulu. Dokter akan memberikan slip resep pada pasien sehingga pasien dapat mengetahui obat apa saja yang akan ditebus beserta nomer pengambilan obat. Dengan sistem *e-prescribing* diharapkan pasien tidak perlu menunggu antrean pengambilan obat terlalu lama. Di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta pembiayaan dokter dan obat-obatan adalah gratis bagi tentara beserta keluarga. Sistem *electronic prescribing* menggunakan *framework codeigniter* karena performanya yang cepat dan menggunakan *pattern MVC* (*Model View Controller*), sehingga struktur *code* menjadi lebih terstruktur dan memiliki standart.

Penulis menentukan beberapa batasan masalah dalam pengerjaan penelitian ini, agar dalam pengerjaan tidak melebar dari rumusan masalah, batasan-batasan tersebut adalah data yang digunakan adalah data obat-obatan penyakit gigi dan mulut, studi kasus Rumah Sakit Gigi dan Mulut TNI-AU Jakarta, dokter hanya menangani pasien yang telah terdaftar, obat racik sepenuhnya adalah tanggung jawab apoteker, hasil akhir berupa aplikasi berbasis web yang dibangun dalam bentuk sistem sebagai pendukung keputusan resep elektronik dengan menggunakan *framework codeigniter*, dan sistem ini hanya dapat digunakan secara *online*. Berdasarkan analisis yang dilakukan penulis terhadap kondisi yang terjadi di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta, maka dapat dirumuskan bagaimana cara untuk mengembangkan sistem *e-prescribing* resep dokter dengan menggunakan *framework codeigniter*. Adapun tujuan Penelitian ini adalah mengembangkan sistem *e-prescribing* resep dokter dengan menggunakan *framework codeigniter*. Manfaat dari Penelitian ini adalah untuk mengubah pencatatan resep dokter dari konvensional menjadi sistem resep elektronik agar lebih efektif dan efisien yaitu dengan mengurangi waktu antrean penebusan obat, menghindari kesalahan dalam pemberian obat, dan data resep dapat disimpan pada *database*.

II. KAJIAN PUSTAKA

A. Resep

Resep adalah permintaan tertulis dari seorang dokter yang diberi izin berdasarkan peraturan perundang-undangan yang berlaku kepada apoteker untuk menyiapkan atau meracik obat serta menyerahkan obat tersebut kepada pasien [6].

B. E-Prescribing (Electronic Prescribing)

E-prescribing adalah resep yang ditransmisikan menggunakan media elektronik untuk menggantikan tulisan tangan, yang menghubungkan berbagai informasi antara dokter, alat pembuat resep elektronik, dan apotek baik secara langsung ataupun tidak langsung [3].

C. Sistem Informasi

Sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan untuk menjalankan operasional perusahaan. Sistem informasi adalah sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan [5].

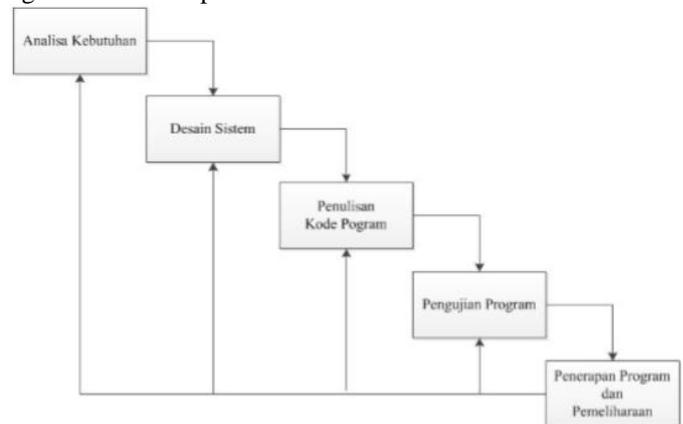
D. System Development Life Cycle (SDLC)

Metode *System Development Life Cycle* atau sering disingkat dengan SDLC merupakan pengembangan yang berfungsi sebagai sebuah mekanisme untuk mengidentifikasi perangkat lunak. Pengembangan sistem informasi yang berbasis komputer merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan, dan

dipelihara. Salah satu model dari metode SDLC adalah model *waterfall*.

1. Model *Waterfall*

Model *waterfall* adalah merupakan model yang paling populer diantara model pendekatan yang lain didalam SDLC dan bekerja secara linier dan berurutan di lain pihak model ini merupakan model yang paling awal terbentuk [1]. Sehingga model *waterfall* mempunyai kelebihan mudah diterapkan dan diaplikasikan serta sesuai apabila digunakan untuk membuat perangkat lunak yang kebutuhannya jelas karena setiap tahapannya diperhitungkan secara maksimal agar kesalahan dapat dihindari.



Gambar 1 : Ilustrasi Model *Waterfall*

E. Unified Modeling Language (UML)

UML adalah bahasa standar industri untuk perancangan perangkat lunak berorientasi objek [8]. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antara *user* dengan *developer*, antara *developer* dengan *developer*, antara *developer* analis dengan *developer* desain, dan antara *developer* desain dengan *developer* pemrograman. UML memungkinkan *developer* melakukan pemodelan secara visual, yaitu penekanan pada penggambaran. Dalam penelitian ini, peneliti menggunakan tiga jenis diagram yaitu *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah apa yang dilakukan sistem bukan bagaimana sistem itu bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

2. *Class Diagram*

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class* memiliki tiga area pokok yaitu nama, atribut, dan metoda. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem dan menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi).

3. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi,

dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

F. ERD (*Entity Relationship Diagram*)

Menurut Marlinda, *Entity Relationship Diagram* (ERD) adalah diagram dari sistem yang menggambarkan hubungan antar entitas beserta relasinya yang saling terhubung [2]. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan dari data *user*.

G. Database

Database adalah suatu koleksi data yang saling berhubungan secara logis dan menggambarkan integrasi antara suatu tabel dengan tabel lainnya, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Pada umumnya data dalam *database* bersifat *integrated* dan *shared*. Maksud dari *integrated* adalah *database* merupakan penggabungan beberapa file data yang berbeda, dengan membatasi pengulangan baik keseluruhan file atau pun sebagian. Pengertian *shared* adalah data individu dalam *database* dapat digunakan secara bersamaan antara beberapa penggunaan yang berbeda [4].

H. MySQL

MySQL adalah sebuah *SQL client/ server relational database management system* yang berasal dari Scandinavia. MySQL termasuk *SQL server*, program *client* untuk mengakses server, hal-hal yang berguna dalam hal administrasi, dan sebuah *programming interface* untuk menulis program sendiri. MySQL dapat berjalan pada personal komputer. Pengembangan dari MySQL terjadi pada sistem yang tidak mahal contohnya (Linux System). Selain itu MySQL juga dapat berjalan secara *portable* pada sistem operasi yang komersial seperti Windows, Solaris, Irix. MySQL menggunakan bahasa SQL. SQL (*Structured Query Language*) adalah bahasa standard yang digunakan untuk mengakses *server database*.

I. Codeigniter (CI)

Codeigniter adalah *framework php* yang berjalan pada *php 4* dan *php 5*. Tujuan utama dari *codeigniter* adalah untuk memudahkan seorang *programmer* dalam mengembangkan aplikasi secara cepat tanpa harus melakukan pemrograman dari nol [2]. *Model View Controller* merupakan suatu konsep yang cukup terkenal dalam pembangunan aplikasi web.

J. Web Server XAMPP

Xampp merupakan sebuah *web server* yang bertindak sebagai *server* yang berdiri sendiri atau *local server (localhost)*, yang terdiri dari beberapa program bawaan antara lain: *Apache HTTP Server*, *MySQL Database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. *Xampp* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP*, dan *Perl*. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman *web* yang dinamis [7].

III. IMPLEMENTASI PROGRAM

A. Tahap Penelitian

Pada penelitian ini, penulis melakukan tiga tahap dalam melaksanakan penelitian yaitu perencanaan, pelaksanaan dan penulisan. Adapun tahap-tahap yang penulis lakukan adalah sebagai berikut :

1. Tahap Perencanaan

Pada tahap perencanaan, penulis melakukan beberapa kegiatan secara terurut yakni penulis memilih tema atau topik penelitian, melaksanakan studi pendahuluan, merumuskan masalah, membuat instrumen penelitian serta menyusun rencana penelitian.

2. Tahap Pelaksanaan Penelitian

Pada tahap ini, penulis mengimplementasikan metode pengembangan sistem model *waterfall*. Tahapan-tahapan sesuai dengan model tersebut adalah analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, serta penerapan program dan pemeliharaan dimana tahap pengujian program akan dibahas pada Bab IV (Uji Coba dan Hasil Uji Coba).

3. Tahap Penulisan

Pada tahap penulisan, penulis membuat sebuah laporan penelitian dengan bentuk dan sistematik skripsi berdasarkan pedoman skripsi jenis rekayasa produk atau aplikasi produk.

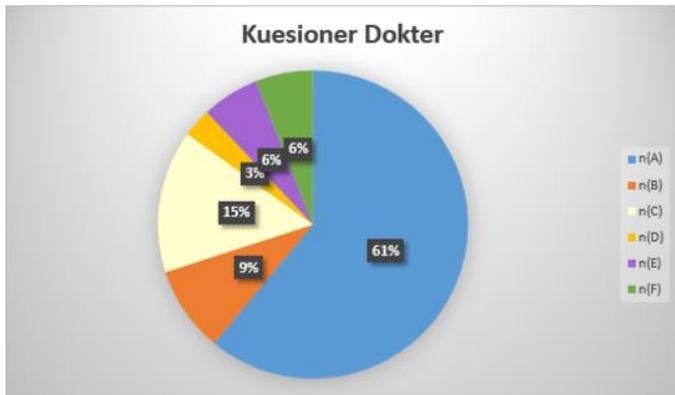
B. Analisa Kebutuhan

Analisa kebutuhan dilakukan dengan mengumpulkan beberapa informasi yang dibutuhkan untuk membangun sistem informasi *electronic prescribing* dengan menyebarkan kuesioner kepada dokter dan apoteker Rumah Sakit Gigi dan Mulut TNI-AU Jakarta untuk mengetahui apa saja fitur yang dibutuhkan pada sistem *electronic prescribing*. Dari hasil kuesioner yang didapat dari dokter Rumah Sakit Gigi dan Mulut TNI-AU Jakarta diperoleh informasi bahwa dibutuhkan suatu sistem *electronic prescribing* yang terintegrasi dengan apotek sehingga pasien sudah mendapatkan nomer pengambilan obat diapotek secara otomatis ketika dokter telah selesai menginput resep.

Pemberian nomer pengambilan obat diapotek secara otomatis diharapkan dapat mengurangi waktu antrean penebusan obat dikarenakan obat sudah dapat mulai diracik bahkan sebelum pasien sampai di apotek. Selain itu dari hasil kuesioner yang didapat dari apoteker Rumah Sakit Gigi dan Mulut TNI-AU Jakarta diperoleh informasi bahwa dibutuhkan sistem yang dapat mencetak resep yang telah diinput oleh dokter, agar pasien dapat mengetahui obat apa saja yang diberikan dibandingkan dengan resep konvensional yang ditulis tangan dengan menggunakan tulisan dokter yang biasanya tidak dapat dibaca oleh pasien.

Kuesioner yang disebarkan juga menanyakan fitur apa saja yang seharusnya terdapat dalam sistem *electronic prescribing* ini. Dari 37 responden, yaitu 33 responden dari dokter dan 4 responden dari apoteker mengatakan fitur-fitur yang diharapkan seperti mengetahui antrean pasien serta melihat stok obat-obatan. Fitur lain yang disarankan responden dokter

adalah fitur untuk melihat riwayat medis pasien, membuat jadwal *appointment*, dan fitur melihat jadwal *appointment*. Sedangkan fitur lain yang disarankan responden apoteker adalah fitur untuk melihat transaksi hari ini, melakukan transaksi pembelian obat bagi pasien yang tidak terdaftar di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta, dan fitur untuk melihat laporan penjualan. Berdasarkan data yang telah terkumpul, hasil kuesioner dari 33 responden dokter adalah sebagai berikut:



Gambar 2 : Grafik Hasil Kuesioner Dokter

Berdasarkan data yang telah terkumpul, hasil kuesioner dari 4 responden apoteker adalah sebagai berikut:



Gambar 3 : Grafik Hasil Kuesioner Apoteker

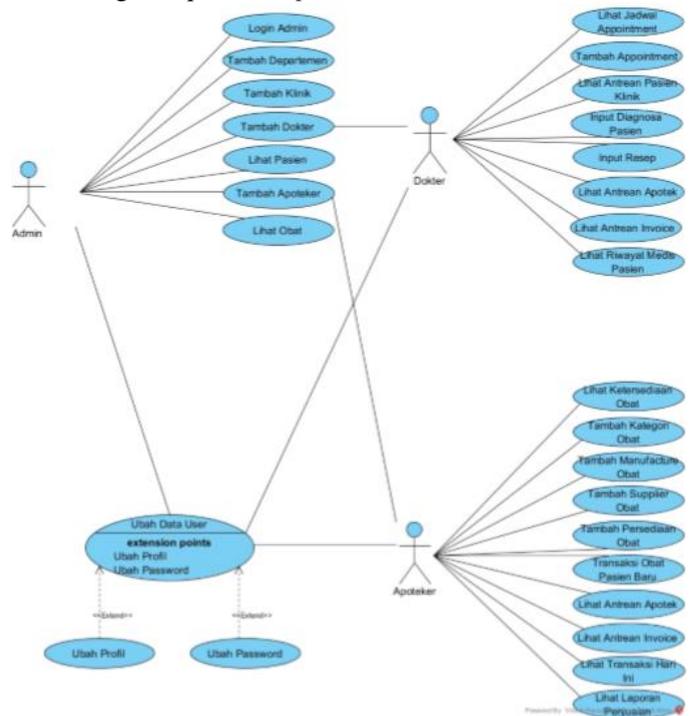
C. Desain Sistem

Pada tahap desain sistem dilakukan perancangan sistem yang dibuat berdasarkan hasil dari analisis kebutuhan yang dilakukan pada tahap sebelumnya dengan membuat diagram UML untuk merancang struktur *database* yang akan digunakan pada sistem. Diagram UML yang akan digunakan adalah *use case*, *class*, dan *activity*. Kemudian akan dibuat desain tampilan sistem yang akan mempermudah user dalam menjalankan sistem. Berikut adalah desain diagram untuk sistem resep elektronik:

1. Use Case Diagram

Pada *use case diagram* akan ditampilkan 3 aktor yang berperan sebagai admin, dokter, dan apoteker. Setiap aktor diwajibkan untuk melakukan login terlebih dahulu

sebelum memasuki sistem. Admin memiliki wewenang untuk mengelola dan menambah data *user* baru agar dapat *login* ke sistem seperti menambah klinik dan menambah apoteker. Selain itu admin juga memiliki kewenangan untuk menambah departemen dan dokter, melihat siapa saja pasien yang terdaftar di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta, dan mengubah profil dan *password*. Dokter memiliki wewenang untuk menambah jadwal *appointment* dengan pasien dan melihat jadwal tersebut, melihat antrian pasien di klinik, menginput diagnosis pasien, memberi resep pasien, melihat antrian apotek, melihat antrian *invoice*, melihat riwayat medis pasien, serta mengubah profil dan *password*. Apoteker memiliki wewenang melihat ketersediaan obat, menambah kategori obat (manufaktur obat, *supplier* obat, dan persediaan obat), melakukan transaksi pemesanan obat oleh pasien baru yang tidak terdaftar, melihat antrian apotek dan antrian *invoice*, melihat transaksi hari ini, melihat laporan penjualan, serta mengubah profil dan *password*.

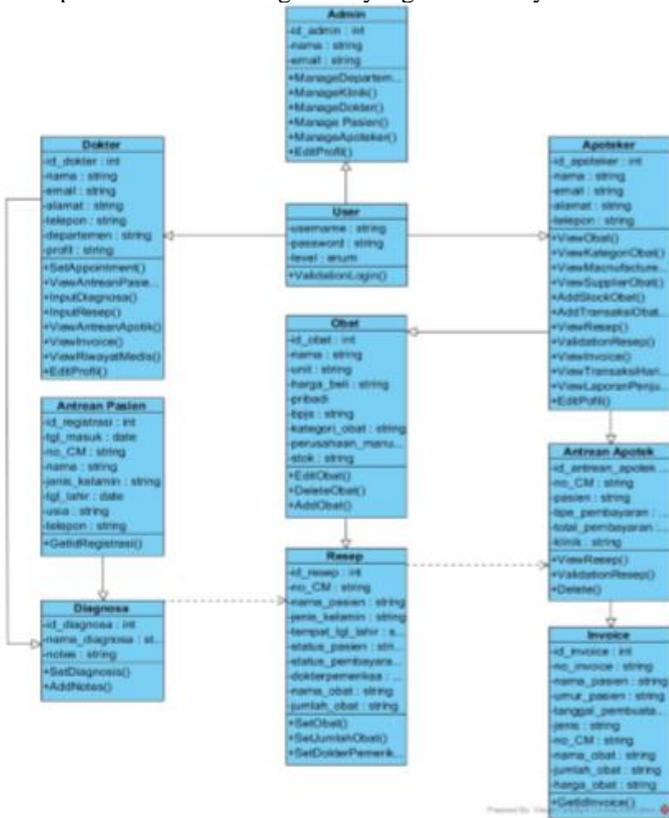


Gambar 4 : Use Case Diagram Sistem Informasi Electronic Prescribing

2. Class Diagram

Class diagram didesain memiliki sepuluh class yaitu *user*, admin, dokter, apoteker, antrian pasien, *diagnose*, obat, resep, antrian apotek, *invoice*. Masing-masing *class* mewakili data-data yang akan diolah sistem. Setiap *class* memiliki *operation* yang berguna untuk mengelola data. *Class user* berfungsi untuk *login* dalam sistem yaitu admin, dokter, dan apoteker. *Class* antrian pasien menyediakan informasi siapa saja pasien yang ingin berobat di klinik tersebut. *Class* diagnosis menyediakan informasi mengenai macam-macam penyakit gigi mulut. *Class* resep menyediakan informasi tentang obat apa saja yang diberikan oleh dokter kepada pasien yang dimana data obat-obatannya bersumber dari *class* obat. Informasi mengenai siapa saja yang akan menebus obat

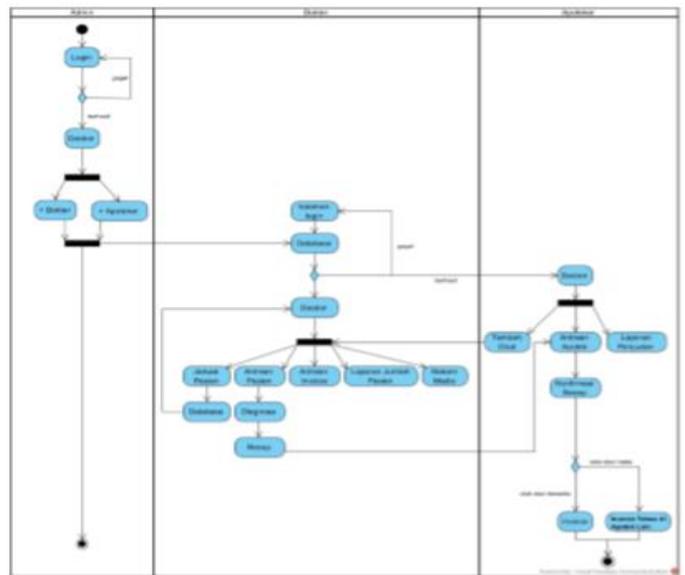
telah disediakan oleh sistem pada *class* antrian apotek. *Class invoice* menyediakan informasi tentang apa saja obat yang telah pasien tebus dan harga obat yang harus dibayar.



Gambar 5 : Class Diagram Sistem Informasi Electronic Prescribing

3. Activity Diagram

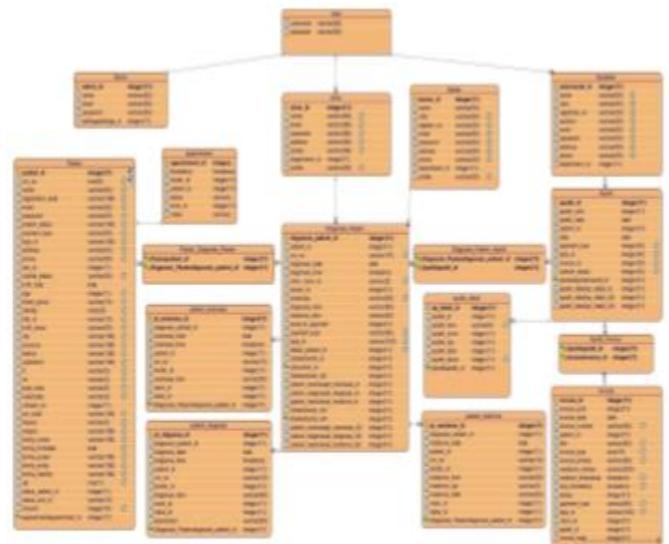
Activity diagram merupakan desain *activity* alur sistem dari setiap *user* yaitu *admin*, dokter, dan apoteker. *Activity diagram* ini menggambarkan alur *electronic prescribing* yang dimulai dengan *admin* yang menambahkan dokter dan apoteker agar dapat *login* ke sistem, kemudian dokter memberi resep kepada pasien lalu apoteker akan memvalidasi resep. *Invoice* diterbitkan apabila validasi resep telah dilakukan oleh apoteker. Apabila obat tidak tersedia di apotek, maka akan diterbitkan *invoice* lain untuk menebus obat di apotek lain.



Gambar 6 : Activity Diagram Sistem Informasi Electronic Prescribing

4. Entity Relation Diagram (ERD)

Penulis menggunakan *Entity Relationship Diagram* untuk menggambarkan bagaimana alur dari penyimpanan data pada sistem *electronic prescribing*. Terdapat 14 tabel yang akan menjadi media penyimpanan data dari sistem. Tabel *user* yaitu *admin*, *clinic*, apoteker berfungsi untuk menyimpan setiap data dari *user*.



Gambar 7 : Entity Relation Diagram Sistem Informasi Electronic Prescribing

5. Desain Tampilan Awal

Pada sistem *electronic prescribing* ini terdapat 3 *user* (pengguna) yaitu *admin*, dokter, dan apoteker. *User* diharuskan *login* terlebih dahulu sebelum memasuki sistem.



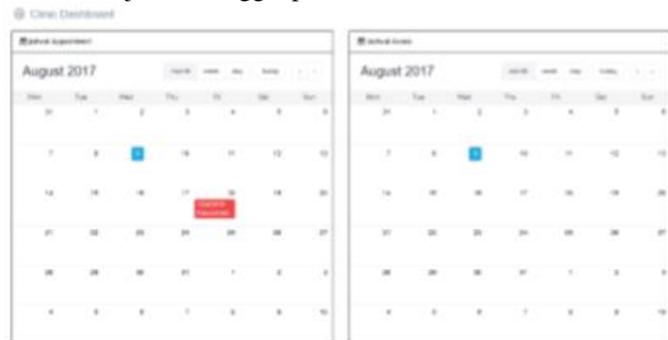
Gambar 8 : Halaman *Login* Sistem Informasi *Electronic Prescribing*

Setelah *user* melakukan *login*, maka *user* akan langsung diarahkan ke tampilan menu *dashboard*. Gambar 9 adalah tampilan *dashboard user* yaitu admin yang menunjukkan jumlah dokter, pasien, apoteker, dan obat-obatan di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta.



Gambar 9 : Halaman *Dashboard Admin*

Gambar 10 menunjukkan tampilan *dashboard user* yaitu dokter yang menampilkan kalender *appointment* pasien. Dokter menginput jadwal *appointment* pasien pada tanggal 18 Agustus 2017 pada pukul 12 siang. Sehingga pada kalender jadwal *appointment* akan muncul label kotak berwarna merah yang berisi jadwal *appointment* pasien. Label kotak berwarna biru menunjukkan tanggal pada hari ini.



Gambar 10 : Halaman *Dashboard Dokter*

Gambar 11 menunjukkan tampilan siapa saja pasien yang akan berobat di klinik.



Gambar 11 : Halaman *Antrean Pasien Klinik*

Pasien yang telah terdaftar pada antrean pasien akan diperiksa dan diberi resep oleh dokter. Dokter akan menginput obat apa

saja yang akan ditebus oleh pasien di apotek ke dalam sistem seperti gambar 12.



Gambar 12 : Halaman *Input Resep Oleh Dokter*

Kemudian pasien akan mendapatkan nomer pengambilan obat di apotek seperti yang terlihat pada gambar 13.



Gambar 13 : Halaman *Nomer Pengambilan Obat Pasien di Apotek*

Gambar 14 adalah tampilan resep yang telah diinput oleh dokter. Pasien tidak perlu memberikan nomer pengambilan obat ke apotek terlebih dahulu karena nomer pengambilan obat sudah masuk secara otomatis dan telah terintegerasi dengan apotek. Pasien dapat mengetahui obat apa saja yang diberikan oleh dokter.



Gambar 14 : Tampilan *Resep dan Nomer Antrean Pengambilan Obat*

D. *Penulisan Kode Program*

Penulisan kode program merupakan tahap untuk menerjemahkan desain sistem yang telah dibuat pada tahap sebelumnya ke bahasa yang dapat dikenali oleh komputer. Bahasa yang digunakan dalam sistem ini adalah HTML, PHP, CSS, Javascript dan JQuery. Pada tahap ini desain sistem yang sebelumnya telah dibuat akan diwujudkan agar menjadi sistem yang berjalan dengan baik sesuai dengan analisis kebutuhan. Tahapan awal dalam penulisan kode program adalah dengan

membuat basis data yang berguna untuk menyimpan data-data yang diperlukan dalam sistem *electronic prescribing*. Tahap selanjutnya adalah implementasi desain awal sistem pada *web*. Setelah tampilan selesai dibuat penulis akan membuat fitur-fitur sistem *electronic prescribing*. Adapun tahapan dalam implementasi program adalah sebagai berikut :

1. Membangun Basis Data (*Database*)

Basis Data dibuat dengan menggunakan MySQL yang pembuatannya menggunakan aplikasi *PhpMyAdmin*. Penulis membuat tabel dalam basis data sesuai dengan ERD yang dibuat sebelumnya.

2. Implementasi Desain Antarmuka Sistem

Desain sistem awal yang sebelumnya dibuat akan diimplementasikan pada tahap ini. Implementasi desain sistem sudah menggunakan *framework codeigniter* sehingga tampilan sudah dapat dilihat pada *web* atau *browser* menggunakan server lokal atau *localhost* dengan *web server XAMPP*. Pada sistem informasi *electronic prescribing*, pertama kali *user* akan melihat halaman *login*. Setelah melakukan proses *login*, *user* akan masuk ke dalam sistem dan akan dialihkan oleh sistem pada halaman *dashboard* dari sistem tergantung pada status *user*.

Pemeriksaan pasien dimulai dari dokter yang *menginput* anamnesa pasien. Anamnesa merupakan pengambilan data oleh dokter dari hasil wawancara dengan pasien mengenai gejala-gejala penyakit yang dirasakan oleh pasien. Setelah melihat keadaan pasien dan melakukan anamnesa, dokter akan menentukan penyakit apa yang sedang dialami oleh pasien dan *menginput* diagnosis tersebut. Dokter akan *menginput* obat apa saja yang akan ditebus oleh pasien agar pasien lekas sembuh. Data-data yang telah *diinput* oleh dokter akan dikirim ke apotek sehingga apotek langsung dapat mengetahui resep yang telah *diinput* oleh dokter. Dokter juga dapat melihat antrean pasien di apotek dan mencetak nomer pengambilan obat pasien agar ditunjukkan pada saat menebus obat di apotek. Apoteker yang telah berhasil login dan *mengclick* sub menu antrean apotek, akan diarahkan ke halaman yang menampilkan siapa saja pasien yang akan menebus obat.

Apoteker akan melakukan konfirmasi terhadap resep yang telah *diinput* oleh dokter. Konfirmasi dilakukan untuk mengecek ketersediaan stok obat diapotek. Apoteker juga berhak untuk menghapus atau mengganti obat yang telah *diinput* oleh dokter dengan catatan obat yang diganti harus mempunyai khasiat yang sama dengan obat yang telah diresepkan. Setelah melakukan konfirmasi maka akan muncul halaman untuk mencetak *invoice* seperti pada gambar 15. *Button* cetak obat habis akan muncul, apabila stok obat yang diresepkan oleh dokter habis. *Button* tersebut digunakan untuk mencetak lembaran yang akan digunakan pasien untuk menebus obat di apotek lain.



Gambar 15 : Halaman Antrean *Invoice*

3. Implementasi Sistem (Pemrograman)

Penulis menggunakan bahasa pemrograman PHP dengan menggunakan *framework codeigniter* dalam mengerjakan proses pemrograman atau pengkodean. Proses pemrograman dibagi kedalam tiga bagian berdasarkan konsep MVC yang diterapkan dalam *framework codeigniter* yaitu bagian untuk mengatur tampilan (*view*), bagian untuk manipulasi *database* (*model*), dan bagian untuk mengatur antara tampilan dan *database* (*controller*). Sistem *electronic prescribing* mempunyai dua fitur utama yaitu fitur peresepan obat pasien oleh dokter dan fitur konfirmasi resep dokter oleh apoteker. Selain fitur-fitur yang telah disebutkan, penulis juga membuat fitur-fitur pendukung seperti fitur *login*, fitur untuk mengatur jadwal *appointment*, dan fitur untuk mencetak resep obat. Implementasi fitur-fitur dalam sistem dimulai dengan fitur *login*. Sistem ini dapat diakses oleh tiga *user* yaitu administrator, dokter, dan apoteker. Masing-masing *user* memiliki tugas yang berbeda. Langkah awal yang penulis lakukan adalah membuat model *login*. Selanjutnya penulis membuat *view* untuk tampilan *login* di sistem agar *user* dapat memasukkan *username* dan *password*. Setelah itu penulis membuat *controller* yang berguna untuk memvalidasi *user* yang akan mengakses sistem.

Fitur-fitur lain pada sistem *electronic prescribing* ini juga menggunakan cara yang sama seperti fitur *login*. Untuk fitur peresepan obat oleh dokter, sistem akan menampilkan halaman dengan form yang akan diisi oleh *user* dokter yaitu dengan *menginput* anamnesa, diagnosis, dan obat. Penulis pertama kali membuat *view* untuk menampilkan form peresepan, kemudian penulis membuat *model* dan *controller* untuk menghubungkan antara *view* dan *model*.

IV. UJI COBA DAN HASIL UJI COBA

A. Uji Coba

Pada metode pengembangan sistem model *waterfall*, setelah melakukan tahap implementasi proses pengembangan dilanjutkan dengan tahap pengujian program. Pada tahap uji coba penulis melakukan pengujian sistem terhadap 3 responden yang akan menjadi pengguna sistem nantinya. Pengujian terhadap sistem dimaksudkan untuk mengetahui apakah sistem yang dikembangkan sudah sesuai dengan kebutuhan dari pengguna. Uji coba sistem dilakukan terhadap 3 responden yaitu administrator, dokter klinik, dan apoteker di Apotek Rumah Sakit Gigi dan Mulut TNI-AU Jakarta. Setiap

responden akan melakukan pengujian terhadap sistem berdasarkan peran dari setiap responden yaitu *user* administrator, user dokter dan user apoteker. Pada tahap uji coba ini, penulis menggunakan kuesioner *user acceptance test* dalam mengumpulkan informasi dari hasil pengujian komponen-komponen dari sistem informasi *electronic prescribing*.

B. Hasil Uji Coba

1. Administrator

Pada bagian uji coba untuk administrator, penulis melakukan pengujian dengan administrator Rumah Sakit Gigi dan Mulut TNI-AU Jakarta. Adapun hasil dari pengujian *user* administrator untuk setiap komponen dapat dilihat pada tabel 1.

TABEL 1 Komponen Penilaian Uji Coba Sistem Untuk Administrator

No.	Fitur	Detail	Keterangan
1	Departemen	Tambah Departemen	[x] Sesuai
		Search Departemen	[x] Sesuai
		Edit Departemen	[x] Sesuai
		Delete Departemen	[x] Sesuai
		Cetak Departemen	[x] Sesuai
2	Clinic	Tambah Clinic	[x] Sesuai
		Search Clinic	[x] Sesuai
		Edit Clinic	[x] Sesuai
		Delete Clinic	[x] Sesuai
		Cetak Clinic	[x] Sesuai
3	Dokter	Tambah Dokter	[x] Sesuai
		Search Dokter	[x] Sesuai
		Edit Dokter	[x] Sesuai
		Delete Dokter	[x] Sesuai
		Cetak Dokter	[x] Sesuai
4	Pasien	Search data pasien	[x] Sesuai
		Edit data pasien	[x] Sesuai
		Delete data pasien	[x] Sesuai
5	Apoteker	Tambah Apoteker	[x] Sesuai
		Search Apoteker	[x] Sesuai
		Edit Apoteker	[x] Sesuai
		Delete Apoteker	[x] Sesuai
		Cetak Apoteker	[x] Sesuai
6	Obat-Obatan	Search data obat	[x] Sesuai
7	Akun	Ubah profil	[x] Sesuai
		Ubah password	[x] Sesuai

2. Dokter

Pada bagian uji coba untuk dokter, penulis melakukan pengujian dengan salah satu dokter di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta. Adapun hasil dari pengujian *user* dokter untuk setiap komponen dapat dilihat pada tabel 2.

TABEL 2 Komponen Penilaian Uji Coba Sistem Untuk Dokter

No.	Fitur	Detail	Keterangan
1	Appointment	Tambah Appointment	[x] Sesuai
		Search Appointment	[x] Sesuai
		Edit Appointment	[x] Sesuai
		Delete Appointment	[x] Sesuai
		Cetak Appointment	[x] Sesuai
2	Antrean Pasien	Search Pasien	[x] Sesuai
		Diagnosa Pasien	[x] Sesuai
3	Antrean Apotek	Cetak Antrean Apotek	[x] Sesuai
4	Antrean Invoice	Cetak Antrean Invoice	[x] Sesuai
5	Riwayat Medis	Search Pasien	[x] Sesuai
		Tampilkan Riwayat Medis Pasien	[x] Sesuai
6	Akun	Ubah profil	[x] Sesuai
		Ubah password	[x] Sesuai

3. Apoteker

Pada bagian uji coba untuk *user* apoteker, penulis melakukan pengujian dengan salah satu apoteker di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta. Adapun hasil dari pengujian apoteker untuk setiap komponen dapat dilihat pada tabel 3.

TABEL 3 Komponen Penilaian Uji Coba Sistem Untuk Apoteker

No.	Fitur	Detail	Keterangan
1	Master Data	Manajemen Obat-Obatan	[x] Sesuai
		Manajemen Kategori Obat	[x] Sesuai
		Manajemen Manufacture	[x] Sesuai
		Manajemen Supplier	[x] Sesuai
		Manajemen Stok Obat	[x] Sesuai
2	Transaksi Baru	Submit Data Pasien Baru	[x] Sesuai
		Input Resep Obat	[x] Sesuai
3	Antrean Apotek	Search Pasien	[x] Sesuai
		View Transaction	[x] Sesuai
		Delete Pasien	[x] Sesuai
		Cetak Antrean Pasien	[x] Sesuai
4	Antrean Invoice	Cetak Antrean Invoice	[x] Sesuai
		Cetak Obat Habis	[x] Sesuai
5	Transaksi Hari Ini	Search Pasien	[x] Sesuai
		Cetak Rekap Transaksi Hari Ini	[x] Sesuai
6	Laporan Penjualan	Search Data Laporan Penjualan	[x] Sesuai
7	Akun	Ubah profil	[x] Sesuai
		Ubah password	[x] Sesuai

C. Penerapan Program dan Pemeliharaan

Pada metode pengembangan perangkat lunak *System Develop Life Cycle* model *waterfall*, langkah selanjutnya setelah tahap pengujian program adalah tahap penerapan program dan pemeliharaan. Setelah sistem berjalan dan beroperasi sebagaimana mestinya maka diperlukan adanya pemeliharaan agar sistem dapat berjalan dengan optimal. Setelah tahap penerapan program atau *deployment*, diperlukan pengawasan terhadap sistem setidaknya dalam rentang waktu 4 bulan untuk memastikan sistem berjalan dengan baik dan memeriksa apakah terdapat *bug* atau kesalahan pada sistem informasi *electronic prescribing* dan diperlukan adanya pengawasan terhadap jalannya sistem yang berkaitan dengan *database*.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan dari hasil implementasi dan uji coba sistem informasi *electronic prescribing*, dapat ditarik kesimpulan sebagai berikut:

- Sistem informasi *electronic prescribing* menggunakan model pengembangan *waterfall* dengan tahapan analisis kebutuhan, desain sistem, penulisan kode program, pengujian program, serta penerapan program dan pemeliharaan.
- Sistem informasi *electronic prescribing* menggunakan *framework codeigniter* yang mempunyai pattern MVC (*Model View Controller*). *Model* adalah bagian untuk memanipulasi *database*, *view* adalah bagian untuk mengatur tampilan, dan *controller* adalah bagian untuk mengatur antara *model* dan *view*.
- Sistem *electronic prescribing* ini diharapkan dapat meningkatkan akses pelayanan kesehatan dan meningkatkan kualitas dan efektifitas dari pelayanan yang diberikan di Rumah Sakit Gigi dan Mulut TNI-AU Jakarta.

B. Saran

Adapun saran dari penulis untuk penggunaan sistem informasi *electronic prescribing* dan penelitian selanjutnya adalah sebagai berikut:

- Pemberian pelatihan teknis kepada *user* yang menggunakan sistem *electronic prescribing* agar *user* mampu mengatasi masalah yang mungkin terjadi. Hal ini dilakukan supaya pengguna sistem tidak terlalu bergantung kepada teknisi luar jika sistem *electronic prescribing* mengalami gangguan kecil.
- Fungsi sistem informasi *electronic prescribing* akan optimal jika seluruh pengguna konsisten dan menggunakan sistem secara kontinu.
- Menambahkan fitur tindakan dokter terhadap pasien sehingga riwayat tindakan dokter terhadap pasien dapat disimpan dalam *database* dan dilihat kembali.
- Menambahkan jam penebusan obat pada slip resep obat yang telah diinput oleh dokter, agar pasien dapat mengetahui pada jam berapa obat dapat ditebus di apotek.
- Rumah sakit yang menggunakan sistem *electronic prescribing* diharapkan mempunyai generator listrik untuk mencegah terhambatnya pelayanan kesehatan masyarakat ketika terjadi pemadaman listrik.

REFERENCES

- [1] Dewanto, I. J., "System Development Life Cycle Dengan Beberapa Pendekatan," 2004.
- [2] Marlinda, L., Sistem Basis Data. Yogyakarta: Andi, 2004
- [3] Puspa Setia Pratiwi, A. L., "E-Prescribing: Studi Kasus Perancangan dan Implementasi Sistem Resep Obat Apotik Klinik," 2013.
- [4] Robby, Owen Kwanentent, F. M. W., "Analisis dan Perancangan Basis Data untuk Mendukung Aplikasi ERP Education pada Bina Nusantara University," 2009.
- [5] Rudi Hermawan, Arief Hidayat, V. G. U., "Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web (Studi Kasus: Yayasan Ganesha Operation Semarang)," 2015.
- [6] Susanti, I., "Identifikasi Medication Error Pada Fase Prescribing, Transcribing, Dan Dispensing Di Depo Farmasi Rawat Inap Penyakit Dalam Gedung Teratai, Instalasi Farmasi RSUP Fatmawati Periode 2013," Skripsi, UIN Syarif Hidayatullah Jakarta, Jakarta, 2013.
- [7] Tim, W. (2016) "Xampp." [Online] <https://en.wikipedia.org/wiki/XAMPP>.
- [8] Zhiqun, W., "A Modelling Approach for Use-cases Model in UML," 2012.