APLIKASI RUMAH KECANTIKAN STUDI KASUS AURAKU SKIN SOLUTION

Penulis : Khairul Anwar

Program Studi D3 Manajemen Informatika, Fakultas Ilmu Terapan, Universitas Telkom

[P1]Sari Dewi Budiwati, S.T.,M.T. , [P2]Toufan Diansyah Tambunan, S.T.,M.T.

khairulanwar162@gmail.com

Abstrak

Di Rumah Kecantikan Auraku Skin Solution terdapat berbagai macam aktifitas seperti reservasi pasien yang akan melakukan perawatan, rekam medis, dan transaksi pembayaran. Selain itu juga pegawai disana diharuskan untuk merekap semua data seperti data transaksi, data stok produk dan data pasien. Permasalahannya adalah resepsionis kesulitan ketika akan mencara kartu pasien yang akan reservasi dan juga dalam pengelolaan data stok produk karena harus direkap setiap harinya. Kemudian bagian kasir juga memerlukan waktu yang tidak sedikit apabila sedang mengelola transaksi karena proses transaksi yang dilakukan hanya menggunakan mesin kasir.

Berdasarkan permasalahan tersebut, auraku skin solution membutuhkan suatu aplikasi yang dapat membantu menyelesaikan kendala pengelolaan reservasi, rekam medis, transaksi dan stok produk supaya lebih efektif. Solusi yang akan ditawarkan untuk membantu permasalahan tersebut yaitu dengan dibuatnya aplikasi berbasis web menggunakan basis data untuk penyimpanan datanya, dan juga dibuatnya fitur-fitur pada setiap fungsionalitas seperti pencarian, penambahan, dan mencetak laporan.

Berdasarkan hasil pengujian *User Acceptance Testing*, 100% pengguna yang terlibat dalam aplikasi ini lebih banyak mengatakan bahwa aplikasi ini sudah Baik dan sudah sesuai dengan kebutuhan masing – masing pengguna dan memahami konsep dari sistem yang dibuat. Maka dari itu dapat disimpulkan bahwa aplikasi ini sudah dapat membantu pegawai yang ada di auraku skin solution dalam mengelola data seperti reservasi, rekam medis, pengelolaan stok produk sampai transaksi.

Kata Kunci : Auraku Skin Solution, Reservasi, Rekam Medis, Stok Produk

Abstract

In the Beauty Auraku Skin Solutions, there is a wide range of activities such as a reservation the patient who will perform maintenance, medical records, and payment transactions. In addition, employees are required to collect all data such as transaction data, data of product and data the patient. The problem is the desk clerk into trouble when it will finding the patients who are going to the reservation and also in the management of data of product because they have to collected every day. Then the cash register also need time not least when it is managing the transaction because the process of transactions made using only the cash register.

Based on the problems, auraku skin of solution requires an application that can help resolve problems of a reservation, medical records, transactions and stock products to be more effective. The solution will be offered to help the problem is with the issue of web based applications using a database for data storage, and also the issue of features on any functionality such as search, additions, and a report.

Based on the results of testing User Acceptance of testing was definitely, 100 % of users involved in the application of this is a lot to say that this application has been right and is in conformity with the needs of each – each user and understand the concept of a system that is made. Therefore, we can conclude that this application has been able to help employees in auraku skin of solution in managing the data like a reservation, medical records, the management of stock products to the transaction.

Keyword : Auraku Skin Solutions, Reservation, Medically, Stock Products

1. Pendahuluan

1.1 Latar Belakang

Rumah Kecantikan Auraku adalah salah satu organisasi bisnis di bidang jasa yang melayani berbagai macam pelayanan perawatan kecantikan. Lokasi rumah kecantikan yang dijadikan studi kasus penulis ini terdapat di jl. Soekarno Hatta. Rumah kecantikan ini beroperasi setiap hari Senin-Sabtu

dari pukul 09:00 – 17:00. Kemudia terdapat 4 bagian pegawai yaitu resepsionis, kasir, dokter, dan terapis. Berdasarkan hasil wawancara, saat ini Rumah Kecantikan Auraku memiliki permasalahan dalam mengelola data seperti reservasi pasien yang dilakukan oleh bagian resepsionis. Resepsionis kesulitan ketika mencari kartu pasien yang akan melakukan reservasi karena kartu pasien disimpan pada lemari arsip. Resepsionis juga kesulitan untuk mengelola data stok produk yang masuk dan stok produk yang terjual karena harus direkap setiap harinya di *Microsoft Excel*. Selanjutnya adalah bagian aktifitas dokter. Dokter akan melakukan pengelolaan data rekam medis seperti hasil pemeriksaan, resep produk dan perawatan yang akan di ambil. Sampai saat ini semua aktifitas pencatatan data tersebut hanya di catat pada kartu pasien.

Kemudian bagian kasir bertugas untuk mengambil resep produk, dan menghitung total pembayaran. Aktifitas ini dilakukan menggunakan mesin kasir dengan cara melihat jenis perawatan ataupun produk yang harus beli yang sudah di catat pada kartu pasien oleh dokter. Bagian kasir harus menghafal kode produk dan perawatan untuk dapat menghitung total biaya tersebut.

Dari semua permasalahan di atas, maka pada proyek akhir ini diusulkan untuk membuat aplikasi yang dapat mengelola data seperti menambah, mengubah dan menghapus. Kemudian semua data yang dikelola akan disimpan pada basis data secara komputerisasi dan saling berintegrasi kemudian menyediakan fitur pencarian berdasarkan data yang dibutuhkan. Diharapkan dapat membuat solusi dalam proses pengelolaan data menjadi lebih efektif, efisien dan mendapatkan data yang akurat.

1.2 Rumusan Masalah

Rumusan masalah pada proyek akhir ini adalah :

1) Bagaimana cara mengelola data reservasi pasien, rekam medis, dan data transaksi tanpa harus menggunakan Microsoft Excel ataupun buku?
2) Bagaimana cara merekap stok produk dan mengetahui tanggal berapa saja di *supply*, tanggal berapa terjual dan berapa sisa stok produk tersedia?
3) Bagaimana cara membantu pegawai untuk menyimpan dan membuat laporan dari data pasien, data produk, dan transaksi agar tidak mudah hilang?
4) Bagaimana cara membantu dokter dalam melihat riwayat rekam medis pasien seperti tanggal rekam medis, hasil periksa dan produk atau perawatan yang dilakukan berdasarkan kode pasien yang melakukan rekam medis?

1.3 Tujuan

Dari rumusan permasalahan diatas, maka Proyek Akhir ini akan diusulkan untuk membangun aplikasi yang :

1) Memiliki fitur aplikasi yang dapat mengelola data reservasi pasien, rekam medis, dan transaksi dengan dibuatnya *form* isian sesuai data yang dibutuhkan.
2) Memiliki fitur untuk menambahkan jumlah *supply* perhari dan memiliki fungsi yang dapat menghitung perbandingan antara stok produk dengan penjualan menggunakan metode *First In First Out (FIFO)*.
3) Membangun aplikasi yang menyediakan fitur cetak laporan data pendaftaran pasien, data produk, dan transaksi dan dapat menyimpan semua data yang sudah diolah kedalam basis data.
4) Memiliki fitur aplikasi untuk pencarian berdasarkan kode pasien pada menu rekam medis dan riwayat rekam medis.

1.4 Batasan Masalah

Batasan pada aplikasi ini adalah :

1) Aplikasi ini hanya di akses menggunakan server lokal, tidak secara online, jadi hanya bisa di akses di dalam Rumah Kecantikan tersebut.
2) Untuk menjalankan aplikasi ini pengguna perlu menambahkan aplikasi pendukung agar aplikasi bisa berjalan dengan lancar.
3) Aplikasi ini membutuhkan satu personal komputer yang aktif selama aktifitas berjalan sebagai komputer server agar komputer lain bisa mengakses aplikasi dengan lancar.

1.5 Definisi Operasional

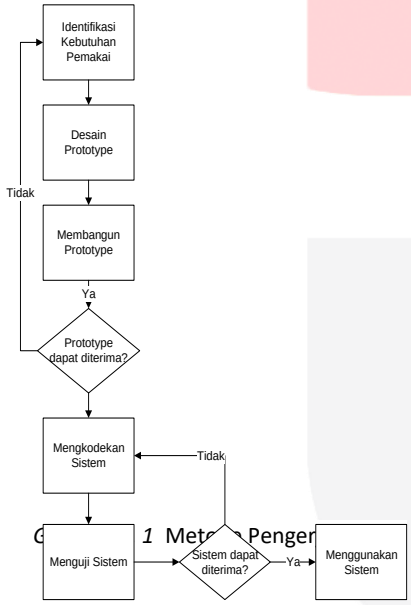
Proyek Akhir ini akan membuat aplikasi berbasis web yang bertujuan untuk memudahkan pegawai dalam melakukan pencatatan data pendaftaran pasien yang sudah menjadi anggota maupun bukan anggota. Kemudian memudahkan dalam pencatatan

data stok produk, dan membantu dokter dalam melakukan rekam medis. Selanjutnya memudahkan bagian kasir untuk dapat melakukan perhitungan biaya perawatan dan pembelian produk. Aplikasi ini juga akan dibuatkan fitur pencarian dan cetak laporan sesuai data yang dibutuhkan pengguna. Aplikasi ini akan dibangun dengan metode prototype.

1.6 Metode Pengerjaan

Dalam metode pengerjaan akan membahas mengenai perencanaan tentang Proyek Akhir yang akan dibangun. Metode yang tepat untuk membuat Proyek Akhir ini adalah metode *prototype.* Terdapat 8 tahapan pada model prototype, yaitu Analisis kebutuhann pemakai, desain *prototype*, membuat *prototype*, persetujuan *prototype*, pengkodean sistem, menguji sistem, persertujuan sistem, menggunakan sistem [2].

1) Identifikasi Kebutuhan Pemakai



G 1 Met Penger

Identifikasi merupakan tahap awal yang dilakukan dalam membangun suatu sistem. Pada tahap ini penulis melakukan identifikasi masalah dan mengumpulkan semua data kebutuhan pengguna sesuai sistem yang akan dibangun. Identifikasi masalah dan pengumpulan data dilakukan dengan cara :

a. Observasi, pada tahap ini penulis mengumpulkan data dengan meninjau secara langsung proses bisnis yang berjalan di Rumah Kecantikan Auraku mengenai sistem pengelolaan data dan transaksinya.

b. Wawancara, pada tahap ini penulis melakukan wawancara secara langsung dengan pemilik usaha Rumah Kecantikan Auraku.

2) Desain *Prototype*

Pada tahap ini akan dibuat desain dari sistem berdasarkan hasil analisis pada proses sebelumnya. Pembuatan desain ini yaitu perancangan menggunakan UML (*Unified Modeling Language*) seperti *Usecase*, *Entity Relationship Diagram(ERD)*, *Flowmap*.

3) Membangun *Prototype*

Setelah desain dibuat, aktifitas selanjutnya adalah membangun *prototype* dari sistem pengelolaan data Rumah Kecantikan Auraku. *Prototype* berupa tampilan *mockup* yang dibuat untuk memudahkan ketika akan mulai pengkodean sistem.

4) Persetujuan *Prototype*

Tahap ini digunakan sebagai evaluasi yang dilakukan oleh user apakah *prototype* yang sudah dibangun sudah sesuai dengan keinginan user. Jika sudah sesuai maka langkah selanjutnya dapat dikerjakan. Jika tidak *prototype* direvisi dengan mengulang dari langkah pertama

5) Pengkodean Sistem

Dalam tahap ini dilakukan penerjemah dari desain model dalam sistem. Aplikasi yang akan dibangun adalah sistem berbasis web dengan menggunakan bahasa pemograman HTML, CSS, JavaScript dan PHP dengan *framework* CI. Sedangkan basis data yang akan digunakan adalah MySQL.

6) Pengujian Sistem

Pada tahap ini dilakukan evaluasi fungsionalitas sistem untuk memastikan fungsionalitas berjalan sesuai yang diharapkan atau tidak. Pengujian menggunakan metode *Black Box Testing* yaitu pengujian yang dilakukan dengan cara mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

7) Persetujuan Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sesuai dengan yang diharapkan . Jika sesuai maka tahap selanjutnya dapat dikerjakan. Jika tidak, tahap 5 diulang kembali.

8) Menggunakan sistem

Pengerjaan Proyek Akhir tidak sampai pada tahap ini.

2. Tinjauan Pustaka

2.1 Profil AURAKU SKIN SOLUTION

Auraku Skin Solution merupakan perusahaan yang bergerak dalam bidang produk dan jasa perawatan kecantikan. Perusahaan ini berdiri sejak tahun 2006 di Kota Bandung. Pusat penjualan produk dan jasa Auraku Skin Solution berada di Jl. Pondok Mutiara dan Jl. Taman Mutiara Cimahi, Bandung. *Auraku Skin Solution* membuka beberapa cabang di Kota Bandung yaitu di Jl. Sukabumi No. 38 Bandung, Jl. Margahayu Raya, Jl. Cipaganti, dan Jl. Soekarno Hatta. Selain di Kota Bandung, *Auraku Skin Solution* membuka cabang di Kota Karawang dan Kota Malang.

Auraku Skin Solution memiliki pegawai seperti resepsionis yang bertugas untuk melayani konsumen yang datang ke auraku, dokter memiliki tugas untuk melakukan pemeriksaan kepada pasien, kasir bertugas untuk melakukan penerimaan transaksi dari konsumen, dan terapis bertugas untuk melakukan tindakan perawatan kepada pasien yang akan melakukan perawatan.[14]

2.2 Usecase

Use Case atau diagram *usecase* merupakan pemodelan untuk sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *usecase* yaitu pendefinisian apa yang disebut aktor dan *usecase*.[9]

- Aktor merupakan orang, proses, atau sistem lain yang beriteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.3 Flow Map

Flowmap adalah diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi dan merupakan diagram alir yang menunjukkan arus dari dokumen, aliran data fisis, entitas sistem informasi dan kegiatan operasi yang berhubungan dengan sistem informasi.

Seperti jumlah orang migrasi, jumlah barang yang diperdagangkan atau jumlah paket dalm jaringan. *Flowmap* membantu analis dan programmer untuk memecahkan masalah kedalam segmen atau bagian yang lebih kecil dan membantu dalam menganalisis alternatif-alternatif dalam pengoperasian.[10]

2.4 Entity Relationship Diagram (ERD)

Pemodelan awal basis data yang paling banyak adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional.

Jenis hubungan diantara dua tipe entitas dinyatakan dengan istilah hubungan *one-to-one*, *one-to-many*, *mane-to-one*, dan *many-to-many*. Diasumsikan terdapat dua buah tipe entitas yang diberi nama A dan B, penjelasan dari masing-masing jenis hubungan tersebut adalah sebagai berikut:

a. Hubungan *one-to-one* (1:1)

   Setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B, begitu pula sebaliknya.

b. Hubungan *one-to-many* (1:M)

   Setiap entitas pada tipe entitas A bisa perpasangan dengan banyak entitas pada tipe entitas B, sedangkan setiap entitas pada entitas B hanya dapat berpasangan dengan satu entitas pada entitas A.

c. Hubungan *many-to-one* (M:1)

   Setiap entitas pada tipe entitas A paling banyak berpasangan dengan satu entitas pada tipe entitas B dan setiap entitas pada tibe entitas B dapat berpasangan dengan banyak entitas pada tipe entitas A.

d. Hubungan *many-to-many* (M:M)

   Setiap entitas pada suatu tipe entitas A bisa berpasangan dengan banyak entitas pada tipe entitas B dan begitu pula sebaliknya. [8]

2.5 UML (*Unified Modeling Language*)

Pada perkembangan teknik pemograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek, yaitu *Unified Modeling Language (UML)*. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi pengguna UML tidak terbatas pada metologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.
Berikut ini penjelasan singkat dari pembagian kategori tersebut,

- *Struture Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

- *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

Interaction Diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsitem pada suatu sistem. [2]

2.6 MySQL (My Structure Query Language)

MySQL adalah sebuah sistem manajemen basis data relasi (*relational database managemen system*) yang bersifat "tebuka" (*open source*). Terbuka maksudnya adalah MySQL boleh di-download oleh siapa saja, baik versi kode program aslinya (*source code program*) maupun versi binernya (*executetable program*) dan bisa digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. Perintah SQL sering juga disebut *Query*. Karena menggunakan bahasa standar yang sama, maka tidak akan menjadi kendala besar bila suatu saat nanti Anda berhubungan dengan basis data selain MySQL [11]

2.7 CodeIgniter

CodeIgniter (selanjutnya disebuat CI) adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun dengan menggunakan bahasa PHP. Di dalam CI ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemograman dalam mengembangkan aplikasinya.

Pemograman dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan. Beberapa fitur yang ada dalam CodeIgniter [4] :

a. Sistem berbasis Model – View – Controller,
b. Benar benar *framework* yang ringan,
c. Memiliki fitur *class* basis data yang mendukung beberapa *platform*,
d. Dukungan basis data dengan *active record*,
e. *Form* validasi data,
f. Keamanan dan *filtering* XSS,
g. Manajemen Sesi (*Session Management*).

2.8 PHP (Hypertext Preprocessor)

PHP (*Hypertext Preprocessor*) ini merupakan bahasa yang hanya dapat berjalan pada *server* yang hasilnya dapat ditampilkan pada klien. *Interpreter* PHP dalam mengeksekusi kode PHP pada sisi *server* (disebut *server-side*) berbeda dengan mesin maya *Java* yang mengeksekusi program pada sisi klien (*client-side*). PHP merupakan bahasa standar yang digunakan dalam dunia *website*. PHP adalah bahasa program yang berbentuk *script* yang diletakan didalam *server* web.

Melihat perkembangan teknologi *website*, PHP digunakan oleh *developer website* hampir diseluruh dunia karena bahasa pemrograman PHP bersifat *Open Source* serta lebih dinamis dan mampu berjalan pada sistem operasi apapun. Hal inilah yang menyebabkan bahasa pemrograman PHP berkembang dengan pesat. Jadi dapat disimpulkan bahwa PHP merupakan bahasa pemrograman *website* yang dapat digunakan untuk desain sebuah web menjadi interaktif dan dinamis sertam mampu dijalankan dengan kecepatan tinggi dalam sebuah server [6]

2.9 HTML (HyperText Markup Language)

HTML atau *Hypertext Markup Language* adalah suatu format data yang digunakan untuk membuat dokumen *hypertext* (teks pada komputer yang memungkinkan user saling mengirimkan informasi (*request-respon*)). Dokumen HTML harus disimpan dengan ekstensi .htm atau .html. HTML memiliki tag-tag HTML dapat menggunakan huruf besar atau huruf kecil, karena tidak *case sensitive* (membedakan huruf besar dan huruf kecil memiliki maksud berbeda).

HTML merupakan bahasa script dasar yang berjalan bersama berbagi bahasa pemograman lainnya. Kode HTML tidak dapat dijadikan sebagai *file executable* program, karena HTML hanyalah sebuah bahasa *Scripting* yang dapat berjalan apabila dijalankan didalam *browser* seperti Internet Exploler, Netscape Navigator, Opera, Mozilla dan lain-lain. [6]

2.10 Black Box Testing

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak yaitu unit, integrasi, fungsional, sistem dan penerimaan.

Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga. Metode ujicoba blackbox memfokuskan pada

keperluan fungsional dari software. Karna itu ujicoba blackbox memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba blackbox bukan merupakan alternatif dari ujicoba whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode whitebox. Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya[12]:

- Fungsi-fungsi yang salah atau hilang.
- Kesalahan interface.
- Kesalahan dalam struktur data atau akses database eksternal.
- Kesalahan performa.
- Kesalahan inisialisasi dan terminasi.

2.11 User Acceptance Testing (UAT)

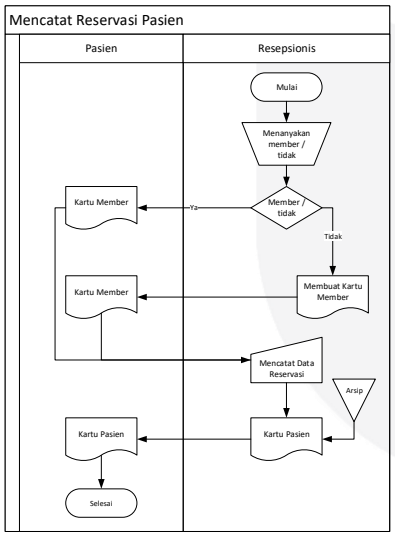
User Acceptance Test (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menhasilkan dokumen yang dijadikan bukti bahwa software yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (testing) sudah bisa dianggap memenuhi kebutuhan dari pengguna. Proses UAT didasarkan pada dokumen requirement yang disepakati bersama.

Dokumen requirement adalah dokumen yang berisi lingkup pekerjaan software yang harus dikembangkan, dengan demikian makan dokumen ini semestinya menjadi acuan untuk pengujian. [13]

3. Analisis dan Perancangan

3.1 Flowmap Sistem Berjalan

3.1.1 Mencatat Reservasi Pasien



Gambar 3. 1

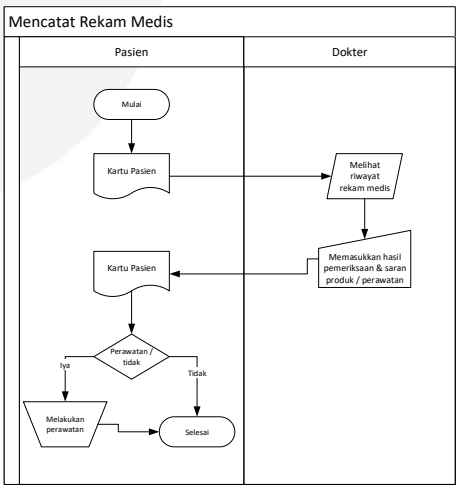
Sistem Berjalan Reservasi Pasien

Reservasi pasien dilakukan ketika pasien akan melakukan perawatan, berikut adalah uraian dari gambar *flowmap* di atas .

a. Resepsionis akan menanyakan apakah pasien *member* atau tidak,
b. Jika ya maka resepsionis akan meminta kartu *member* dari pasien,
c. Jika tidak maka resepsionis akan membuatkan kartu *member* dan memberikannya ke pasien,
d. Resepsionis akan mencatat data reservasi pasien,
e. Mencari kartu pasien dari arsip,
f. Memberikan kartu pasien kepada pasien.

Kelemahan dari sistem yang berjalan saat ini adalah akan sulitnya resepsionis untuk mencari kartu pasien yang sudah menjadi *member*.

3.1.2 Mencatat Rekam Medis



Gambar 3. 2

Sistem Berjalan Rekam Medis

Proses rekam medis dilakukan ketika pasien sudah melakukan pendaftaran pasien, berikut adalah uraian dari gambar *flowmap* diatas.

a. Pasien memberikan kartu pasien kepada dokter,

b. Dokter akan melihat riwayat rekam medis bila diperlukan.

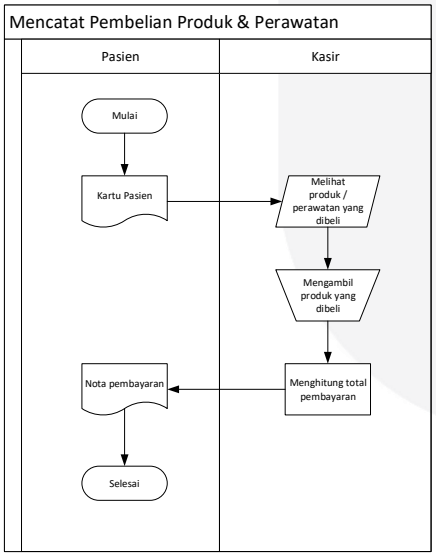
c. Dokter akan mencatat hasil pemeriksaan, dan akan memberikan saran produk dan saran perawatan pada kartu pasien,

d. Kartu pasien diberikan kembali pada pasien,

e. Jika ada perawatan yang akan dilakukan maka pasien akan melakukan perawatan, Jika tidak maka selesai

Kelemahan pada sistem berjalan saat ini adalah sulitnya dokter ketika ingin melihat riwayat rekam medis sesuai keingin dokter, karna semua data riwayat rekam medis tercatat pada kartu pasien. Apabila dokter ingin memberikan saran produk yang harus dibeli, dokter tidak mengetahui jumlah stok produk yang tersedia karena hanya resepsionis dan kasir yang memiliki data stok produk.

3.1.3 Mencatat Pembelian Produk & Perawatan



Gambar 3. 3

Sistem Berjalan Transaksi Pembelian

Transaksi pembayaran dilakukan ketika pasien sudah selesai melakukan perawatan dan pembelian produk. Berikut adalah uraian dari gambar *flowmap* di atas.

a. Pasien memberikan kartu pasien kepada kasir,

b. Kasir akan melihat perawatan dilakukan dan produk yang akan dibeli, dan menghitung total yang harus dibayar oleh pasien,

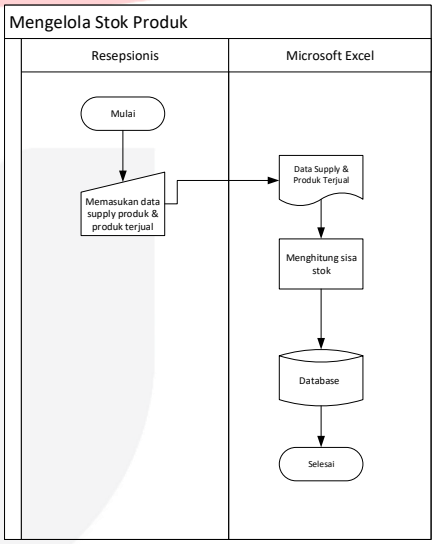
c. Kasir mengambil produk yang akan dibeli,

d. Kasir akan menghitung total pembayaran,

e. Memberikan nota pembayaran pada pasien

Kelemahan pada sistem berjalan saat ini adalah kasir perlu melihat kembali obat yang harus dibeli yang memungkinkan tulisan tidak terbaca. Kasir perlu menghafal kode produk untuk proses perhitungan pembayaran. Karena perhitungan pembayaran dilakukan menggunakan mesin kasir.

3.1.4 Mengelola Stok Produk



Gambar 3. 4

Sistem Berjalan Kelola Stok Produk

Kelola stok produk dilakukan ketika ada produk masuk dan produk terjual. Berikut adalah uraian dari gambar *flowmap* di atas.

a. Resepsionis memasukkan data produk yang di *supply* dan terjual,

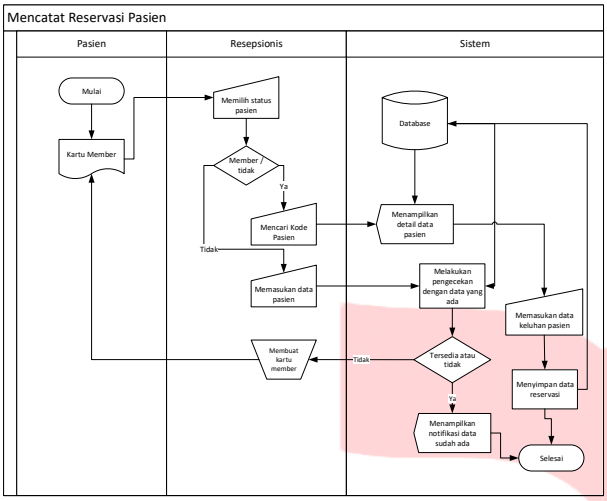
b. Sistem menghitung sisa produk,

c. Menyimpan ke basis data.

Kelemahan dari sistem berjalan saat ini adalah, resepsionis harus merekap ulang jumlah produk yang terjual setiap harinya.

3.2 Flowmap Sistem Usulan

3.2.1 Mencatat Reservasi Pasien



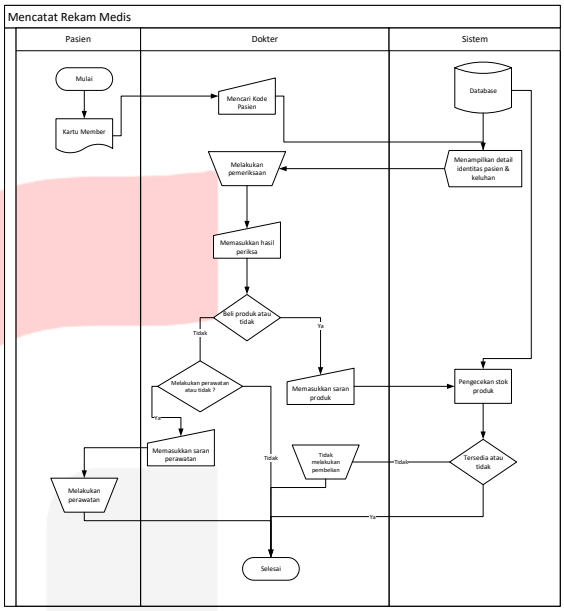
Gambar 3. 5
Sistem Usulan Reservasi Pasien

Resepsionis memiliki aktifitas untuk mengelola data reservasi pasien, berikut adalah uraian dari *flowmap*.

a. Pasien memberikan kartu member kepada resepsionis,
b. Resepsionis akan memilih status pasien, jika pasien member maka akan langsung mencari kode pasien,
c. Jika bukan maka resepsionis akan mencatat data pasien,
d. Sistem akan mengecek apakah data yang dimasukkan sudah tersedia di basis data atau belum. Jika sudah ada maka akan menampilkan notifikasi bahwa data sudah ada, Jika belum maka akan dibuatkan kartu member oleh resepsionis,
e. Sistem akan menampilkan data identitas pasien setelah resepsionis memasukkan kode pasien,

   Resepsionis akan memasukkan data keluhan pasien dan sistem akan menyimpan data reservasi.

3.2.2 Mencatat Rekam Medis



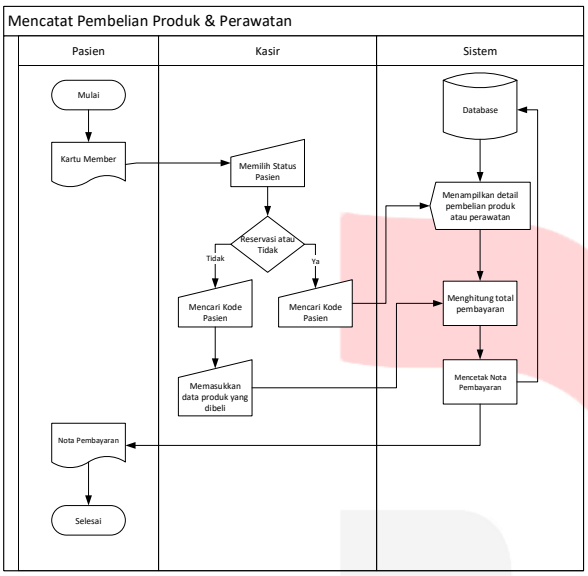
Gambar 3. 6
Sistem Usulan Mencatat Rekam Medis

Mencatat rekam medis dilakukan ketika pasien sudah melakukan reservasi pasien. Berikut adalah uraian dari gambar *flowmap*.

a. Pasien memberikan kartu member kepada dokter,
b. Dokter akan mencari kode pasien,
c. Sistem akan menampilkan identitas pasien yang diambil dari basis data,
d. Dokter akan melakukan pemeriksaan,
e. Dokter akan memasukkan hasil pemeriksaan,
f. Jika pasien perlu melakukan perawatan, maka dokter akan memasukkan saran perawatan. Jika tidak maka dokter tidak perlu memasukkan saran perawatan,
g. Jika pasien perlu membeli produk, maka dokter akan memasukkan saran produk yang harus dibeli. Jika tidak maka dokter tidak perlu memasukkan saran produk,

Sistem akan mengecek apakah produk yang akan dibeli tersedia stoknya atau tidak. Jika tersedia maka membeli produk, Jika tidak tersedia maka batal untuk memberikan saran produk

3.2.2 Mencatat Pembelian Produk & Perawatan



Gambar 3. 7
Sistem Usulan Pembelian Produk atau Perawatan

Mencatat pembelian produk atau perawatan dilakukan ketika pasien sudah melakukan rekam medis ataupun ketika akan melakukan pembelian produk. Berikut adalah uraian dari *flowmap*.

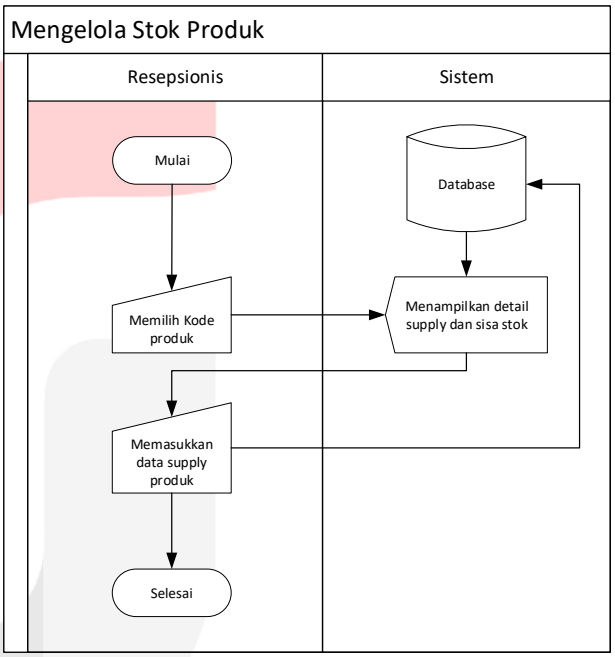
a. Pasien memberikan kartu member pada kasir,

b. Kasir akan memilih status reservasi pasien, jika melakukan reservasi maka hanya perlu memasukkan kode pasien dan sistem akan menampilkan detail transaksi yang diambil dari basis data,

c. Jika tidak melakukan reservasi, kasir akan memasukkan kode pasien dan akan memasukkan data produk yang akan dibeli,

d. Sistem akan menghitung total pembayaran, mencetak nota dan tersimpan di basis data.

3.2.3 Mengelola Stok Produk



Gambar 3. 8
Sistem Usulan Mengelola Stok Produk

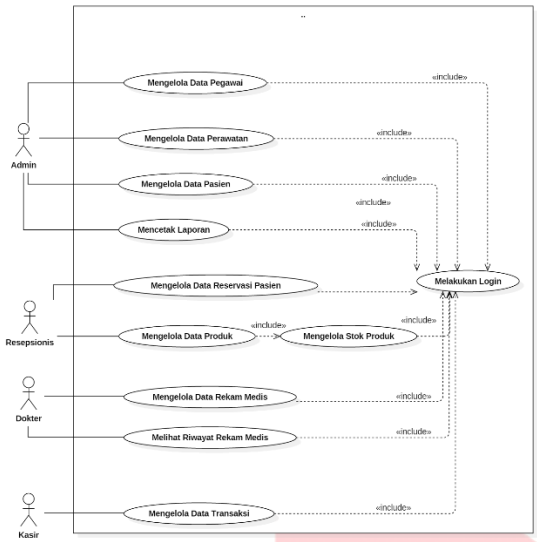
Aktifitas mengelola stok produk dilakukan ketika ada produk yang di *supply*. Berikut adalah uraian dari *flowmap*.

a. Resepsionis memilih kode produk yang di*supply*.

b. Sistem menampilkan detail data *supply* dan sisa stok,

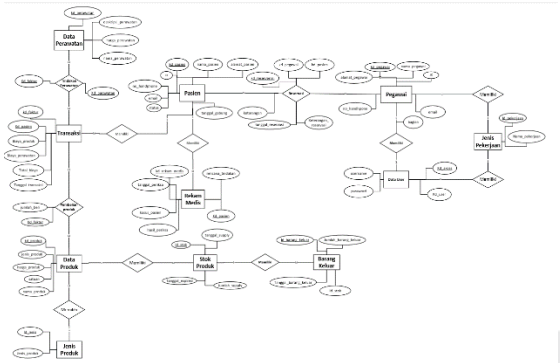
c. Resepsionis memasukkan data *supply* produk

3.2 Usecase Diagram



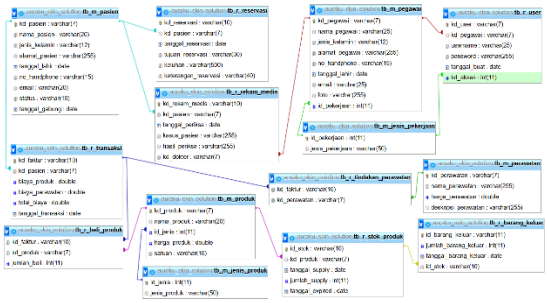
Gambar 3. 9
Usecase Diagram

3.3 Entity Relationship Diagram (ERD)



Gambar 3. 10
ERD

3.4 Skema Relasi



Gambar 3. 11
Skema Relasi

4 Penutup

4.1 Kesimpulan

Dari perancangan, pengujian, dan implementasi aplikasi maka dapat disimpulkan :

1. Aplikasi ini dapat membantu bagian Resepsionis dalam proses reservasi pasien yang akan melakukan perawatan maupun konsultasi dengan mudah dan cepat, karena data sudah terintegrasi dengan basis data dan akan memudahkan resepsionis dalam proses pencarian maupun pengelolaan data pasien,
2. Aplikasi ini juga dapat membantu resepsionis untuk dapat merekap data stok produk yang masuk ataupun keluar, tanpa harus merekap ulang jumlah produk dari hasil penjualan,
3. Aplikasi ini dapat membantu bagian kasir dalam menghitung transaksi yang sudah dilakukan oleh pasien dan membantu merekap hasil transaksi untuk dijadikan laporan transaksi,
4. Aplikasi ini dapat membantu dokter dalam proses perekapan data hasil rekam medis dan juga dapat memberikan resep produk maupun perawatan yang harus dilakukan oleh pasien dan membantu dokter untuk melihat riwayat rekam medis sesuai kebutuhan,

Dari hasil kalkulasi kuesioner 4 responden, 100% responden mengatakan fitur aplikasi sudah dapat membantu kegiatan di auraku dan sudah dapat diterapkan di auraku. Responden juga tidak mengalami kesulitan ketika menggunakan aplikasi tersebut karena sudah paham dengan konsep sistem yang dibuat.

4.2 Saran

Adapun saran yang diajukan aplikasi ini adalah.

1. Aplikasi ini diharapkan dapat dikembangkan menjadi aplikasi online dan terhubung pada semua cabang Auraku Skin Solution
2. Aplikasi ini terdapat data email pada tabel pasien, diharapkan dapat dikembangkan untuk dibuatnya fitur kirim hasil pemeriksaan setiap transaksi
3. Aplikasi ini diharapkan dapat ditambahkan menu fitur baru seperti konsultasi online maupun aplikasi *mobile* untuk konsumen agar mendapatkan info baru dari rumah kecantikan Auraku Skin Solution

Daftar Pustaka

[1] Yusuf dan A. Giriyana, APLIKASI BERBASIS WEB UNTUK PENDAFTARAN PASIEN, PENGELOLAAN DATA PERSEDIAAN OBAT DAN PEMBAYARAN (STUDI KASUS: PRAKTIK DR. DWIYANA M.S), Bandung: FAKULTAS ILMU TERAPAN, MANAJEMEN INFORMATIKA, 2012.

[2] Yulianto dan A. Agung, Analisis dan Desain Sistem Informasi, Bandung: Politeknik Telkom, 2009.

[3] Riyanto, Membuat Sendiri Aplikasi E-Commerce dengan PHP dan MYSQL Menggunakan CodeIgniter dan JQuery, Yogyakarta: ANDI, 2011.

[4] Roger. R. Pressman, Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu), Yogyakarta: ANDI, 2002.

[5] Bunafit. Nugroho, PHP dan MySQL dengan Editor Dreamweaver MX, Yogyakarta: ANDI, 2004.

[6] Angga. A. Ilyas, PERANCANGAN SISTEM INFORMASI REKAM MEDIS PASIEN PADA RUMAH SAKIT DENGAN MENGGUNAKAN RFID (RADIO FREQUENCY IDENTIFICATION), Bandung: Universitas Telkom, 2008.

[7] Arbie, Manajemen Database dengan MySQL, Yogyakarta: ANDI, 2004.

[8] Sutana. E, Basis Data dalam Tinjauan Konseptual, Yogyakarta: ANDI, 2011.

[9] M. Shalahuddin dan Rossa. A.S, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, Bandung: Informatika Bandung, 2013.

[10] M. Shalahuddin dan Rossa. A.S, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berbasis Objek), Bandung: Informatika Bandung, 2011.

[11] M. Rudyanto. Arief, Pemograman Web Dinamis Menggunakan PHP & MySQL, Yogyakarta: ANDI, 2011.

[12] Suhendar. A. S.Si dan Hariman. G. S.Si, Visual Modelling Menggunakan Raltional Rose, Bandung: Informatika, 2001.

[13] Raharjo. Belajar Pemograman Web, Bandung: Modula, 2001.