

RANCANG BANGUN BASIS DATA UNTUK MENAMPILKAN JADWAL PRAKTIK DOKTER MENGGUNAKAN METODE WATERFALL

Ade Sitti Nur Zainab¹, Shasi Aprilia Widiyani², Salmawati³, Astiti Dwi Cahya Ningrum⁴, Tanti Julianingsih⁵,
Rusliani⁶, Natalis Ransi S.Si.,M.Cs.⁷

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo

Jl. HEA Mokodompit, Anduonohu, Kendari–Sulawesi Tenggara

E-mail: adesitti98@gmail.com¹, Lia24.apriliah@gmail.com², salmawati2812@gmail.com³,
astitidwicahyaningrum@gmail.com⁴, tantijulianingsih@gmail.com⁵, rusliani1998@gmail.com⁶,
natalis.ransi@gmail.com⁷

ABSTRAKS

Perancangan basis data merupakan tahapan-tahapan penting pada pembangunan atau pembuatan sebuah sistem informasi. pada makalah ini kami melaporkan rancang bangun basis data jadwal praktik dokter dimana secara keseluruhan kami menggunakan metode waterfall untuk membangun sistem informasi basis data jadwal praktik dokter. Selanjutnya, pada analisis dan perancangan basis data digunakan pendekatan- pendekatan entiti relathionship diagram dalam menentukan entitas-entitas dasar dan nilai kardinalitasnya. Pada pengujian diperoleh bahwa kasus yang telah ditentukan dapat diselesaikan menggunakan pendekatan-pendekatan aljabar relathional yang telah di implementasikan dalam bentuk sql statement, Rancangan basis data yang kami buat juga telah diimplementasikan pada sistem nformasi jadwal praktik dokter. Pada hasil pengujian sistem informasi jadwal praktik dokter dengan menggunakan metode black box diperoleh bahwa fungsionalitas setiap menu berjalan dengan baik/ bernilai 100%.

Kata kunci : system informasi, aljabar relational, metode waterfall.

1. PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi ini, komputer merupakan suatu perangkat yang sangat dibutuhkan untuk proses pengolahan data, agar data yang diolah tersebut dapat secara efektif dan efisien dalam memberikan informasi yang diperlukan oleh suatu perusahaan atau instansi. pengolahan data secara manual, akan membutuhkan waktu dan tenaga yang banyak dan juga informasi-informasi yang diperlukan kurang efisien. Saat ini banyak klinik praktek dokter yang menggunakan system manual dalam pengolahan data serta masih kesulitan dalam membagi waktu untuk melayani pasien. Hal ini dapat menurunkan mutu dan kualitas dari pelayanan yang diberikan oleh klinik terhadap pasien. Sementara dengan aktivitas pasien serta volume barang

inventari yang cukup banyak ,telah menimbulkan berbagai masalah dari pelayanan sehari-hari terutama dalam melayani pasien,inventari obat, dan keuangan.

Berdasarkan latar belakang itulah ,maka perlu adanya system informasi penjadwalan praktik dokter dengan mengedepankan efisiensi dari segi biaya,tenaga dan waktu.Untuk meningkatkan kinerja dalam hal pelayanan terhadap pasien,tantunya membutuhkan sebuah sistem berbasis komputerisasi guna mengatasi masalah-masalah yang ada. Metode yang digunakan untuk merancang sistem ini adalah metode Waterfall dan database MySQL. Sistem informasi ini akan menangani beberapa proses diantaranya registrasi pasien, penjadwalan pemeriksaan pasien hingga pembuatan resep. Adapun tujuan rancang bangun system informasi penjadwalan praktik dokter adalah menampilkan data pasien, data dokter, dan jadwal praktik dokter hingga pembuatan resep serta untuk meningkatkan mutu pelayanan terhadap pasien dan meningkatkan kinerja klinik menjadi lebih baik lagi.

1.2 Landasan Teori

1.2.1 Sistem Informasi

Menurut Dermawan dan Fauzi (2013,13) “Sistem informasi merupakan kumpulan dari sub-sub sistem yang saling berhubungan satu sama lain, yang bekerja sama secara harmonis untuk mencapai satu tujuan, yaitu mengolah data menjadi informasi yang berguna”. Sub-sub sistem tersebut merupakan pengelompokan dari beberapa komponen yang lebih kecil, bagaimana mereka berkelompok bergantung pada interpretasi mereka. Di dalam suatu sistem informasi kalau salah satu unsur tidak ada maka system informasi tersebut tidak akan terwujud terlepas dari bagaimana pengelompokan tersebut dilakukan. Menurut Dermawan dan Fauzi (2013:13) komponen-komponen system informasi sebagai berikut :

- a) Perangkat keras (*Hardware*)
- b) Perangkat lunak (*Software*)
- c) Manusia (*Brainware*)
- d) Prosedur (*Procedure*)
- e) Basis data (*Database*)
- f) Jaringan komunikasi (*Communication network*)

1.2.2 Website

Menurut Fathansyah (2012,464) menyimpulkan bahwa “*World Wide Web* (WWW atau web) merupakan sistem informasi terdistribusi yang berbasis *hypertext*”. Dokumen-dokumen yang dikelola dalam *web* bisa beraneka jenis (pengolah kata, lembar kerja, tabel basis data, presentasi, *hypertext*, dan lain-lain) dan beragam format (.doc, .pdf, .xls, .dbf, .ppt, .html, dan lain-lain). Menurut Kadir (2013:5) “*website* adalah halaman informasi yang ada di internet, dimana halaman tersebut merupakan kumpulan komponen yang terdiri dari teks, gambar atau suara animasi”.

1.2.3 *Mysql Workbench*

MySQL Workbench adalah sebuah perangkat aplikasi berbentuk visual yang dipergunakan untuk mengelola basis data. Perangkat aplikasi ini biasa digunakan oleh seorang arsitek basis data, pengembang basis data, serta administrator basis data. MySQL Workbench menyediakan model data, pengembangan SQL, dan peralatan administrasi yang komprehensif untuk konfigurasi server basis data, administrasi pengguna, dan masih banyak lagi. MySQL Workbench tersedia pada platform Windows, Linux dan Mac OS.

1.2.4 *Personal Home Page* (PHP)

Menurut Anhar (2010,3), “PHP (*PHP Hypertext Preprocessor*) yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*)”. PHP adalah script yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/ *up to date*. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan.

1.2.5 *Entity Relationship Diagram*

a) Definisi ERD

Menurut Al-Bahra (2005:84) “*Entity Relationship Diagram* (ERD) adalah diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis”. *Entity Relationship Diagram* (ERD) merupakan gambaran data yang dimodelkan dalam suatu diagram yang digunakan untuk mendokumentasikan data dengan cara menentukan apa saja yang terdapat tiap entity dan bagaimana hubungan antara entity satu dengan lainnya.

b) Komponen Penyusun ERD

1. Entitas (*Entity*) Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan. Pada ERD, Entitas digambarkan dengan sebuah bentuk persegi panjang. Contoh: Tempat (ruang, bangunan, kantor, lapangan, kampus)
2. Relasi (*Relationship*) Hubungan yang terjadi antara satu entitas atau lebih. Relasi diberi nama dengan kata kerja dasar. Sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif). Pada ERD, Relasi digambarkan dengan sebuah bentuk belah ketupat. Contoh: pasien melakukan registrasi pasien.
3. Derajat Relasi (*Relationship Degree*) *Relationship degree* atau derajat relasi adalah jumlah entitas yang berpartisipasi dalam satu relasi. Derajat relasi yang sering dipakai di dalam ERD:
 - a. *Unary Relationship* *Unary Relationship* adalah model relasi yang terjadi di antara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *recursive relationship* atau *relective relationship*.
 - b. *Binary Relationship* *Binary Relationship* adalah model relasi antar instansi-instansi (*instances*) dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.
 - c. *Ternary Relationship* *Ternary Relationship* merupakan *relationship* antara instansiinstansi dari tiga tipe entitas secara sepihak. Masing-masing entitas mungkin berpartisipasi satu atau banyak dalam suatu *relationship ternary*. Perlu dicatat bahwa *relationship ternary* tidak sama dengan tiga *relationship binary*.
 - d. Atribut (*Attribute*) Atribut merupakan karakteristik dari entitas atau *Relationship* yang menyediakan penjelasan detail entitas atau *relation*. Ada dua jenis atribut:
 1. *Identifier (key)*, digunakan untuk menentukan suatu entitas secara unik (*primary key*).
 2. *Descriptor (nonkey attribute)* digunakan untuk menspesikasikan karakteristik dari suatu entitas yang tidak unik. Pada umumnya atribut merupakan karakteristik dari entitas untuk relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Contoh: Atribut Pelanggan, dan No KTP/SIM, dan Nama
 3. Kardinalitas (*Cardinality*) Menurut Simarmata (2007:98) “Kardinalitas suatu hubungan menyatakan sejumlah kejadian terkait untuk masing-masing dua entitas”.

Kardinalitas mendefinisikan jumlah kemunculan baik minimum maupun maksimum satu entitas yang dapat dihubungkan dengan kemunculan tunggal entitas lain. Jenis Derajat Kardinalitas:

- *One to One* (1:1) Hubungan satu-ke-satu (1:1) terjadi jika sebanyak satu kejadian dari suatu entitas A dihubungkan dengan satu kejadian entitas B.
- *One to Many / Many to One* (1: N / N : 1) Tingkat hubungan satu ke banyak (1:N) adalah sama dengan banyak-ke-satu (N:1). Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas A dapat mempunyai banyak hubungan dengan kejadian pada entitas B. Sebaliknya satu kejadian pada entitas B hanya dapat mempunyai satu hubungan.
- *Many to Many* (M : N) Hubungan banyak-ke-banyak (M:N) mempunyai arti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan banyak entitas pada himpunan entitas A.

1.2.6 Migrasi Data

Migrasi data adalah istilah ilmu komputer yang digunakan untuk proses memindahkan atau mentransformasikan data dari suatu konteks ke konteks lainnya yang berbeda. Konteks tersebut dapat berupa bentuk / struktur data, format data, platform teknologi, ataupun lokasi.

Beberapa alasan utama migrasi data biasanya dilakukan adalah sebagai berikut:

- Basis data yang lama tidak didukung oleh prinsipal utama aplikasi basis data tersebut.
- Buruknya dukungan yang diberikan oleh ekosistem aplikasi tersebut
- Sistem baru yang dikembangkan mengharuskan pemakaian sistem basis data lain.
- Teknologi yang digunakan sudah terlalu lama / usang.

Dilihat dari sisi bisnis maka analisis biaya manfaat (cost benefit analysis) sudah menetapkan jika biaya yang akan ditanggung lebih besar daripada manfaat yang didapatkan jika mempertahankan sistem lama.

Beberapa strategi best practice migrasi data dapat dilakukan dengan cara berikut :

- Dilakukan secara bertahap dan parallel. Parallel disini dalam arti aplikasi yang menggunakan sistem basis data lama tetap dipertahankan sampai sistem pendukung basis data baru dapat menjalankan operasionalnya dengan baik.

- Utilitas / mekanisme teknisnya dapat menggunakan :
 - Aplikasi impor / ekspor data yang biasanya terdapat pada sistem database baru.
 - Scripting / programming dengan bahasa pemrograman populer seperti C#, Perl, PHP, Java, dan lain-lain.
 - Aplikasi khusus ETL (Extract, Transform, and Load) yang biasanya dirancang dengan kemudahan penggunaan antarmuka grafis serta mendukung hampir semua format file maupun sistem basis data populer lainnya.

Khusus untuk penggunaan ETL adopsinya sangat populer belakangan ini. Ini seiring dengan perkembangan pesat metode dan aplikasi Data Warehouse dan Business Intelligence di dunia bisnis. Dengan penggunaan ETL, maka produktivitas migrasi data akan meningkat dan relatif berujung kepada penghematan waktu dan biaya proses migrasi data itu sendiri.

1.3 Metode Penelitian

1.3.1 Aljabar relational

Aljabar relasional adalah bagian dari ilmu komputer, cabang dari logika predikat tingkat pertama dan aljabar himpunan, yang menangani suatu set relasi hingga yang memiliki sifat tertutupan dengan operator-operator tertentu. Operator ini bertindak dengan satu atau lebih relasi untuk menghasilkan suatu relasi (gabungan). Aljabar relasional mulai mendapat perhatian dengan diterbitkannya model relasional data oleh Edgar F. Codd pada tahun 1970 yang mengusulkan untuk menggunakan aljabar ini sebagai dasar dari bahasa kueri basis data.

Kesimpulannya yaitu sekumpulan operasi yang digunakan untuk melakukan proses manipulasi data dalam rangka untuk mendapatkan informasi yang diperlukan dari database. Relasi dalam bahasan ini dipergunakan untuk penamaan tabel beserta datanya baik yang murni maupun yang sudah dilakukan modifikasi dengan operasi-operasi aljabar relasional. Aljabar Relasional di bagi menjadi :

- Operasi *Selection*

Operasi *selection* dapat didefinisikan sebagai "Kumpulan semua tuple-tuple/record-record dalam suatu tabel yang memenuhi kondisi P". Operasi selection berfungsi untuk menyeleksi tuple-tuple yang memenuhi predikat yang diberikan dari sebuah tabel relasi. Simbol sigma " σ " digunakan untuk menunjukkan operasi *select*. Predikat muncul

sebagai *subscript* dari σ dan kondisi yang diinginkan yang ditulis dalam predikat. Argumen diberikan dalam tanda kurung yang mengikuti σ dan berisi tabel relasi yang dimaksud.

- *Operasi Projection*

Operasi *projection* berfungsi untuk memilih nilai atribut-atribut tertentu saja dari sebuah tabel relasi. Simbol π digunakan untuk menunjukkan operasi *projection*. Predikat muncul sebagai *subscript* dari π dan hanya nama atribut yang diinginkan yang ditulis dalam predikat. Argumen diberikan dalam tanda kurung yang mengikuti π dan berisi tabel relasi yang dimaksud.

- *Operasi union*

Operasi *union* berfungsi untuk mendapatkan gabungan nilai atribut dari sebuah tabel relasi dengan nilai atribut dari tabel relasi lainnya. Simbol \cup digunakan untuk menunjukkan operasi *union*. Operasi *union* bernilai benar bila terpenuhi 2 kondisi, yaitu : Derajat dari 2 tabel relasi yang dioperasikan harus sama dan domain dari atribut yang dioperasikan juga harus sama.

- *Operasi Set Difference*

Operasi *set difference* berfungsi untuk mendapatkan nilai yang ada dalam sebuah tabel relasi, tapi tidak ada dalam tabel relasi lainnya. Simbol $-$ digunakan untuk menunjukkan operasi *set difference*.

- *Operasi Cartesian Product*

Operasi *cartesian product* berfungsi untuk mengkombinasikan informasi yang ada dalam 2 tabel relasi dan menghasilkan sebuah tabel relasi yang baru. Simbol \times digunakan untuk menunjukkan operasi *cartesian product*.

- *Operasi Intersection*

Set intersection / Intersection (\cap) termasuk ke dalam operator tambahan, karena operator ini dapat diderivikasi dari operator dasar seperti berikut:

$$A \cap B = A - (A - B), \text{ atau } A \cap B = B - (B - A)$$

Operasi *set intersection* berfungsi untuk mendapatkan nilai yang ada dalam sebuah tabel relasi dan juga ada dalam tabel relasi lainnya. Simbol \cap digunakan untuk menunjukkan operasi *set intersection*.

- *Operasi Natural Join*

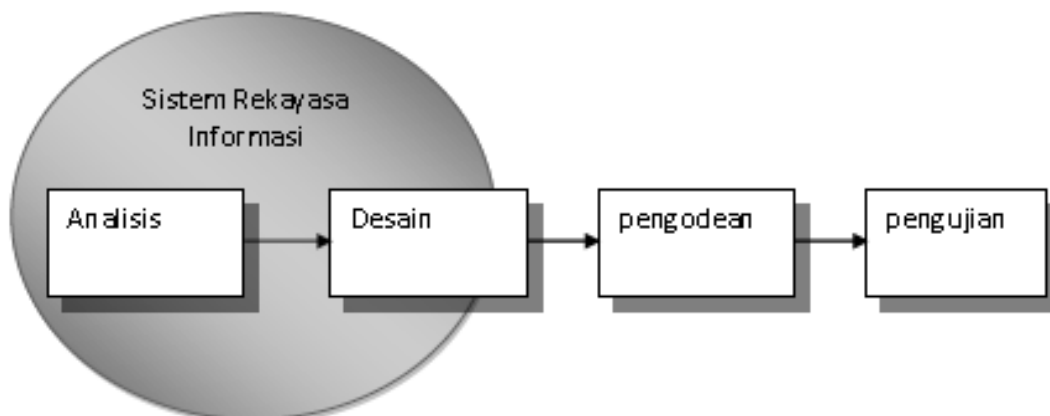
Operasi *natural join* berfungsi untuk menggabungkan operasi *selection* dan *cartesian product* menjadi hanya 1 operasi saja. Simbol “ \bowtie ” digunakan untuk menunjukkan operasi *natural join*. Operasi *natural join* hanya menghasilkan tupel yang mempunyai nilai yang sama pada 2 atribut yang bernama sama pada 2 tabel relasi yang berbeda.

- Operasi *theta join*

Operasi *theta join* berfungsi untuk mengkombinasikan tupel dari 2 tabel relasi dimana kondisi dari kombinasi tersebut tidak hanya nilai dari 2 atribut bernama sama, tetapi kondisi yang diinginkan juga bisa menggunakan operator relasional (\leq , $<$, $=$, $>$, \geq). Operasi *theta join* merupakan ekstensi dari *natural join*.

1.3.2 Metode Waterfall

Sistem informasi yang baik adalah system informasi yang dapat dengan mudah dikembangkan sesuai dengan kondisi dan pengembangan dimana sistem informasi tersebut di aplikasikan, model *waterfall* adalah model yang paling banyak digunakan untuk tahap pengembangan. Model waterfall ini juga dikenal dengan nama model tradisional atau model klasik. Menurut Rosa dan Salahudin (2013:28) mengemukakan bahwa “Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic cycle*)”. Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*). Berikut adalah gambar model air terjun :



Sumber: Rossa dan Salahudin (2013:28)

Gambar 1. Ilustrasi Model *Waterfall*

1. Analisa Kebutuhan

Tahapan ini sangat menekan pada masalah pengumpulan data-data yang dibutuhkan dalam pembuatan web agar dapat merancang konsep serta antarmuka yang dapat menghubungkan lingkungan sekitar.

2. Desain

Proses perancangan sistem ini difokuskan pada empat atribut, yaitu struktur data, representasi antarmuka, arsitektur perangkat lunak, dan interaksi antar objek di dalam lingkungan rumah sakit.

3. Pembuatan Kode Program

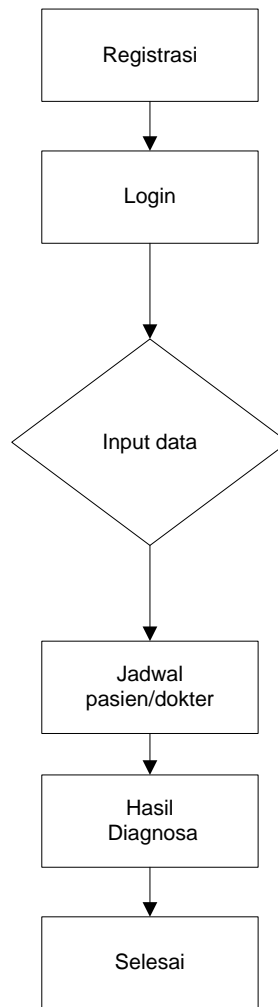
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program.

4. Pengujian

Pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya dengan menggunakan *blackbox testing*.

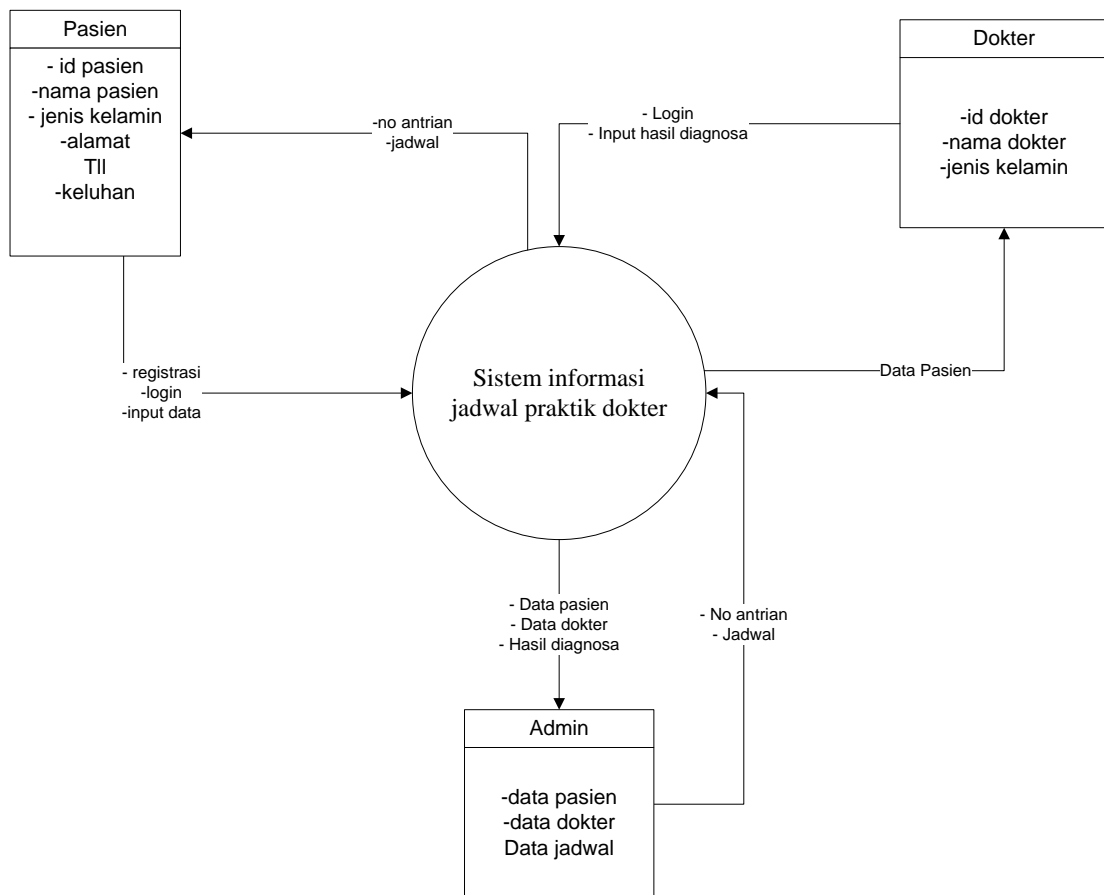
2. PEMBAHASAN

2.1 Flowchart Sistem informasi jadwal praktik dokter



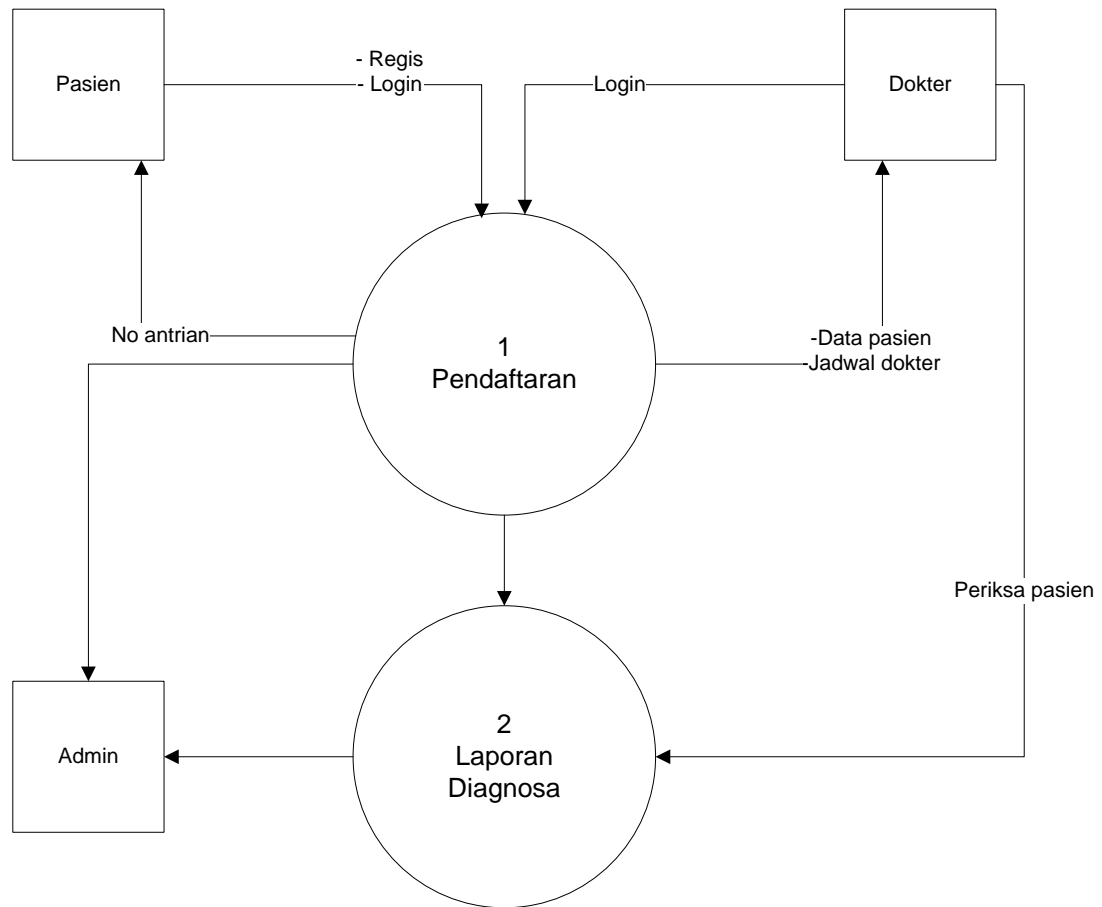
Gambar 2 flowchart system informasi jadwal dokter

2.2 DFD system informasi jadwal dokter



Gambar 3 DFD level-0 system informasi jadwal dokter

2.3 DFD Level-1 system informasi jadwal dokter

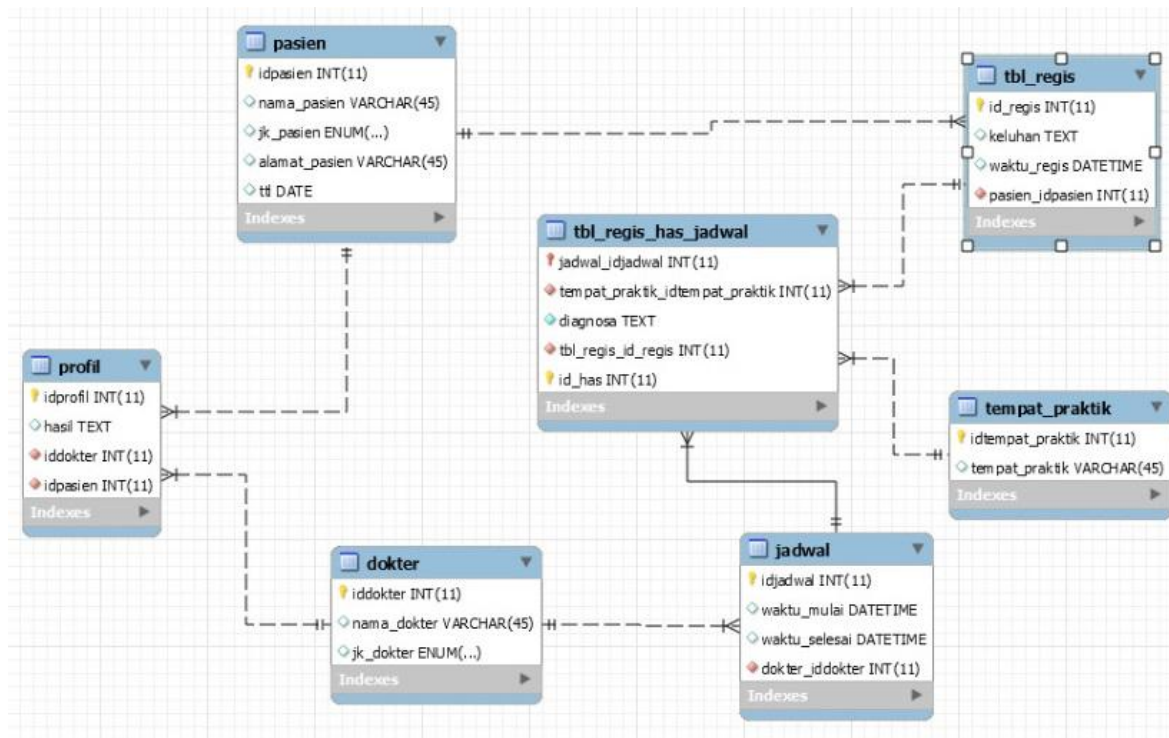


Gambar 4 dfd level-1 system informasi jadwal dokter

Pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya dengan menggunakan *black box testing*. Menurut Romeo (2003) black box testing dilakukan tanpa pengetahuan detail struktur internal system atau komponen yang dites ,sehingga black box testing berfokus pada kebutuhan fungsional pada software berdasarkan spesifikasi kebutuhan dari software :

1. Analisis dan Perancangan

a) Entity Relationship Diagram ERD sistem yang akan dibangun dapat dilihat pada Gambar



Gambar 5. Entity Relationship Diagram

- b) Perancangan Basis Data Dalam ERD diketahui hubungan kardinalitas antar himpunan entitas. Himpunan relasi antar himpunan entitas tersebut ditransformasi ke dalam bentuk tabel sebagai berikut :

Tabel 1. registrasi

No	Elemen Data	Nama Field	Tipe	Pajang	Ket
1.	Kode Regis	id_regis	INT	11	Primary Key
2.	Nomor Antrian	No_antrian	INT	11	
3.	Keluhan Pasien	Keluhan	TEXT		
4.	Waktu Registrasi	Waktu_regis	DATETIME		

Tabel 2. pasien

No	Elemen Data	Nama Field	Tipe	Pajang	Ket
1.	Kode Pasien	idpasien	Int	11	Primary Key
2.	Nama Pasien	nama_pasien	varchar	45	
3.	Jenis Kelamin pasien	jk_pasien	Enum	11	
4.	Alamat Pasien	alamat_pasien	varchar	45	
5.	Tempat Tanggal Lahir Pasien	ttl	Date		

Tabel 3. dokter

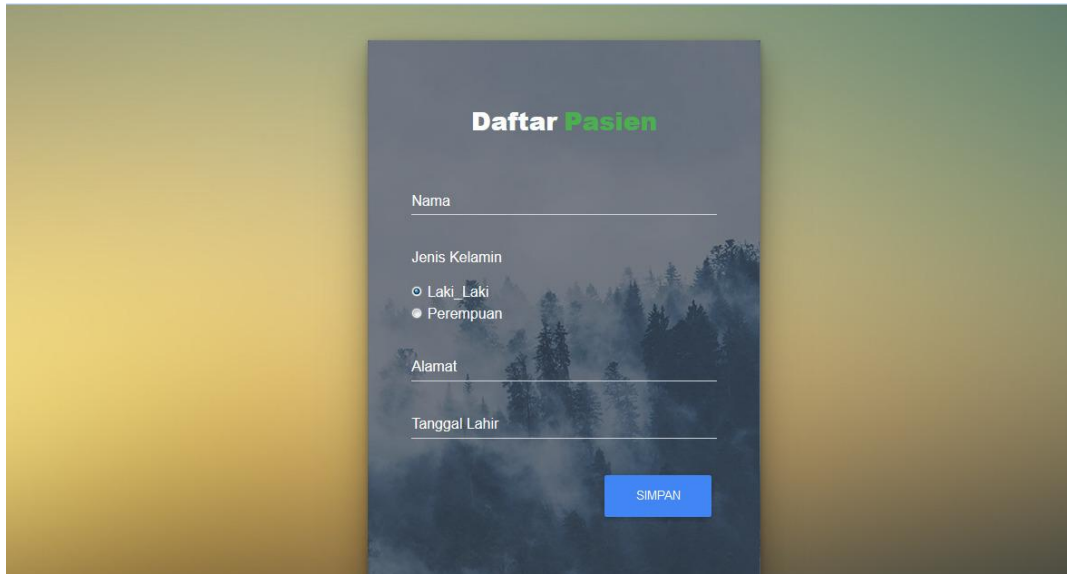
No	Elemen Data	Nama Field	Tipe	Pajang	Ket
1.	Kode Dokter	iddokter	int	11	Primary Key
2.	Nama Dokter	nama_dokter	varchar	45	
3.	Jenis Kelamin Dokter	jk_dokter	Enum	11	

Tabel 4. jadwal Dokter

No	Elemen Data	Nama Field	Tipe	Pajang	Ket
1.	Kode Jadwal	idjadwal	int	11	Primary Key
2.	Waktu Mulai	waktu_mulai	datetime		
3.	Waktu Selesai	waktu_selesai	datetime		

- c) Rancangan Antar Muka (*Interface*) Rancangan antar muka (*interface*) adalah sebuah tampilan rancangan yang akan terlihat dilayar monitor oleh pengunjung web (*user*). Dalam perancangan ini penulis mengembangkan beberapa halaman diantaranya :

- Halaman registrasi



Daftar Pasien

Nama _____

Jenis Kelamin
☐ Laki_Laki
☒ Perempuan

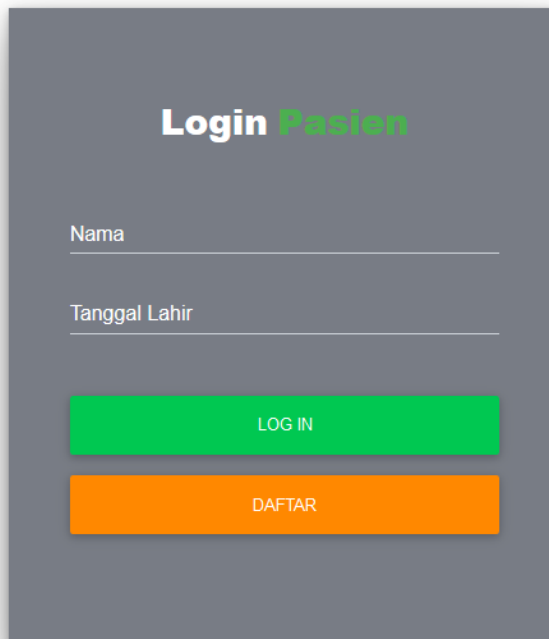
Alamat _____

Tanggal Lahir _____

SIMPAN

(Gambar 6 Halaman reigistrasi)

- Halaman Login Pasien



Login Pasien

Nama _____

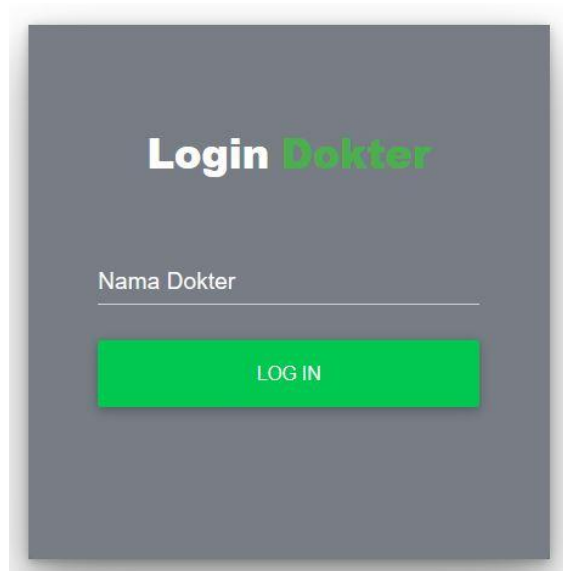
Tanggal Lahir _____

LOG IN

DAFTAR

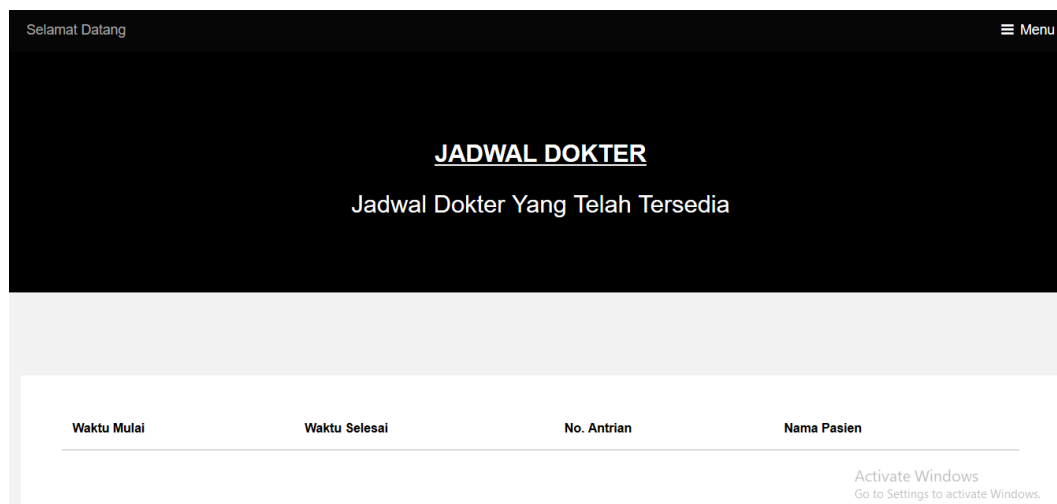
(Gambar 7 Halaman Login Pasien)

- Halaman Login Dokter



(Gambar 8 Halaman Login Dokter)

- Halaman jadwal dokter



(Gambar 9 Halaman jadwal Dokter)

2. Pengujian

No	Scenario Pengujian	Test Case	Hasil	Status
1	Login Dokter	Nama (diisi)	Masuk pada beranda dokter	Valid
2	Registrasi Pasien	Nama (diisi) Jenis Kelamin (diisi) Alamat (diisi) Tanggal Lahir (diisi)	Masuk pada form login pasien	Valid
3	Login Pasien	Nama (diisi) Tanggal Lahir (diisi)	Masuk pada beranda pasien	Valid
4	Ambil Antrian	Tanggal (diisi) Keluhan (diisi) No. Antrian (diisi)	Masuk pada tabel antrian dan pada tabel dokter	Valid
5	Input data Dokter	Nama (diisi) Jenis Kelamin (diisi)	Masuk pada halaman admin	Valid
6	Hapus Data	Klik tombol hapus	Data terhapus	Valid
7	Diagnosa Dokter	Diagnosa (diisi)	Data masuk pada database untuk di tampilkan pada tabel data pasien	Valid
8	Jadwal Dokter	Waktu No. antrian Nama pasien	Menampilkan jadwal dokter yang telah terisi	valid
9	Penjadwalan pasien	Waktu mulai periksa	Menginputkan waktu periksa untuk pasien	valid

3. KESIMPULAN

Perancangan dan pengaplikasian system informasi jadwal praktik dokter terhadap pelayanan pasien pada klinik berbasis web,

1. Adanya *website* ini dapat membantu klinik dalam meningkatkan pelayanan pada pasien di klinik tersebut.
2. Aplikasi system informasi jadwal praktik dokter menyajikan laporan-laporan diantaranya laporan data pasien, laporan data dokter, laporan data tindakan, laporan jadwal dokter,

laporan pendaftaran pasien, dan laporan resep/ obat. Laporan tersebut ditujukan kepada manajer klinik sebagai bahan pertimbangan dalam perbaikan pelayanannya.

3. Adanya *web* ini juga dapat menjadi solusi pelayanan bagi para pasien

4. DAFTAR PUSTKA

- [1] Dermawan, Deni., dan Kunkun Nur Fauzi. 2013. *Sistem Informasi Manajemen*. Bandung: Remaja Rosdakarya.
- [2] Fathansyah. 2012. *Basis Data*. Bandung: Informatika.
- [3] Kadir, Abdul. 2013. *Pemrograman Database MySQL untuk pemula*. Yogyakarta: Mediakom.
- [4] Anhar. 2010. *Panduan Menguasai PHP dan Mysql*. Jakarta: Media Kita.
- [5] Ladjamudin, bin Al-Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- [6] Simarmata, Janner. 2007. *Perancangan Basis Data*. Yogyakarta: Andi.
- [7] A.S, Rosa., dan Shalahuddin M. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- [8] Romeo. 2003. *Testing Dan Implementasi Sistem*. Surabaya: STIKOM.