

**USULAN TUGAS AKHIR  
RANCANG BANGUN SISTEM INFORMASI KONSULTASI  
DAN PENJADWALAN FISIOTERAPI PENYAKIT BELL'S  
PALSY (STUDI KASUS KLINIK GRIYA FISIOTERAPIS)**



**Oleh:**

**REJALANA BAYU KURNIAWAN  
FID 016074**

**PROGRAM STUDI TEKNIK  
INFORMATIKA FAKULTAS TEKNIK  
UNIVERSITAS MATARAM**

**2020**

## USULAN TUGAS AKHIR

### RANCANG BANGUN SISTEM INFORMASI KONSULTASI DAN PENJADWALAN FISIOTERAPI PENYAKIT BELL'S PALSY (STUDI KASUS KLINIK GRIYA FISIOTERAPIS)

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama



**Nadiyahari Agitha, S.Kom., M.MT**  
**NIP. 19860813 201803 2 001**

Tanggal: 14 Mei 2020

2. Pembimbing Pendamping



**Royana Afwani, ST., MT.**  
**NIP. 19850707 201404 2 001**

Tanggal: 14 Mei 2020

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



**Prof. Dr.Eng. I Gede Pasek Suta Wijaya, ST., MT.**  
**NIP: 19731130 200003 1 001**

**USULAN TUGAS AKHIR**  
**RANCANG BANGUN SISTEM INFORMASI KONSULTASI DAN**  
**PENJADWALAN FISIOTERAPI PENYAKIT BELL'S PALSY (STUDI**  
**KASUS KLINIK GRIYA FISIOTERAPIS)**

Oleh:

**Rejalana Bayu Kurniawan**  
**F1D 016 074**

Susunan Tim Penguji

1. Penguji I



**Moh. Ali Albar, ST., M.Eng.**  
**NIP: 19831125 201504 1 002**

Tanggal: 24 Juni 2020

2. Penguji II



**Ir. Sri Endang Anjarwani, M.Kom.**  
**NIP: 19660403 200604 2 001**

Tanggal: 24 Juni 2020

3. Penguji III



**Prof. Dr.Eng. I Gede Pasek Suta Wijaya, ST., MT.**  
**NIP: 19660403 200604 2 001**

Tanggal: 24 Juni 2020



**Prof. Dr.Eng. I Gede Pasek Suta Wijaya, ST., MT.**  
**NIP: 19731130 200003 1 001**

## DAFTAR ISI

HALAMAN COVER.....	0
LEMBAR PENGESAHAN .....	i
DAFTAR ISI.....	iii
ABSTRAK.....	v
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori.....	7
2.2.1 Sistem.....	7
2.2.2 Karakteristik Sistem.....	8
2.2.3 Sistem Informasi .....	9
2.2.4 <i>Website</i> .....	10
2.2.5 <i>Framework Codeigniter</i> .....	10
2.2.6 <i>Bootstrap</i> .....	10
2.2.7 MySQL.....	10
2.2.8 <i>Entity Relationship Diagram</i> .....	11
2.2.9 UML.....	12
2.2.10 <i>Relational Unified Process</i> .....	17
2.2.11 Penyakit Bell's palsy.....	18
BAB III METODE PENELITIAN .....	20
3.1 Diagram Alir Pembuatan Sistem.....	20
3.2 Pengumpulan data .....	20
3.2.1 Observasi.....	21
3.2.2 Wawancara.....	21

3.3 Studi literatur .....	21
3.4 Pengembangan sistem dengan metode RUP .....	22
3.4.1 Tahap <i>Inception</i> .....	22
3.4.2 Tahap Elaboration .....	28
3.4.3 Tahap <i>construction</i> .....	50
3.4.4 Tahap <i>transition</i> .....	50
3.4.5 Jadwal Kegiatan .....	51
DAFTAR PUSTAKA .....	53
LAMPIRAN.....	55

## ABSTRAK

Pada Klinik Griya Fisioterapi, penjadwalan fisioterapi masih menggunakan informasi konsultasi dengan aplikasi WhatsApp dan informasi bagi rekam medik belum tersimpan dengan baik atau hanya mengandalkan ingatan Fisioterapis saja. Pada penelitian ini, dirancang dan dibangun Sistem Informasi dan Konsultasi Penjadwalan Penyakit Bell's Palsy yang dapat membantu dalam mengatasi penjadwalan fisioterapi sekaligus pencatatan rekam medis, sehingga dapat terkontrol dan terkordinir dengan baik. Metode pengembangan dalam sistem informasi yang dirancang menggunakan metode RUP (*Relational Unified Process*). RUP menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak, dimana RUP melakukan pendekatan perangkat lunak secara berulang-ulang (*iterative*), fokus pada arsitektur, lebih diarahkan berdasarkan pengerjaan perangkat lunak mulai dari pengumpulan awal dan penentuan kebutuhan sampai proses pembuatan program. Sehingga cocok untuk perancangan sistem informasi yang akan dibuat dalam penelitian yang diambil.

**Kata Kunci:** Metode *Rational Unified Process*, RUP, Sistem Informasi, Klinik Griya Fisioterapis

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Seiring dengan perkembangan teknologi informasi yang semakin pesat di berbagai bidang, pelayanan kesehatan dituntut untuk melakukan perbaikan secara terus menerus untuk dapat meningkatkan kenyamanan, kemudahan dan keamanan bagi pengguna maupun pengelola jasa kesehatan. Salah satu contoh pelayanan kesehatan pada era modern saat ini adalah pelayanan fisioterapi Bell's palsy.

Fisioterapi merupakan proses merehabilitasi seorang agar terhindar dari cacat fisik melalui serangkaian pencegahan, diagnosis dan penanganan gangguan fisik pada tubuh atau penyakit. Sedangkan Bell's palsy merupakan suatu kelumpuhan akut *nervus facialis prifer* yang tidak diketahui sebabnya [1]. Di Indonesia, insiden penyakit Bell' palsy banyak terjadi namun secara pasti sulit ditentukan. Dalam hal ini didapatkan frekuensi terjadinya Bell's palsy di Indonesia sebesar 19,55%, dari seluruh kasus *neuropati* terbanyak yang sering dijumpai terjadi pada usia 20 – 50 tahun dimana angka kejadian meningkat dengan bertambahnya usia setelah 60 tahun. Maka diperlukannya fisioterapis yang mampu melakukan diagnosis berdasarkan ilmu pengetahuan dan pengalaman yang dimiliki dengan melakukan terapi [2].

Masalah kecacatan yang ditimbulkan oleh Bell's palsy cukup kompleks, yaitu meliputi *impairment* (kelainan di tingkat organ) berupa ketidak simetrisnya wajah, kaku dan bahkan bisa berakibat terjadi kontraktur, disability atau ketidakmampuan berupa keterbatasan dalam aktivitas sehari-hari berupa gangguan makan dan minum, menutup mata, serta gangguan berbicara dan ekspresi wajah, masalah selanjutnya dari segi kejiwaan penderita [3]. Terapi pada kasus Bell's Palsy dengan *Neuro Muscular Taping* (NMT) meningkatkan sirkulasi darah, drainase limfatik, menstimulasi kulit, mendukung otot yang melemah secara kronis, dan meningkatkan kemampuan *motoric*, dan *mirror exercise* merupakan latihan wajah pada sisi yang lesi (abnormal pada tubuh) di depan cermin yang dapat mencegah *atrofi* otot sehingga dapat meningkatkan fungsi otot serta mempercepat proses kesembuhan pasien kondisi Bell's palsy [4].

Pada Klinik Griya Fisioterapi terdapat salah satu fisioterapis yang penting untuk penyembuhan penyakit Bell's palsy di Lombok Nusa Tenggara Barat, namun terdapat kendala dalam penanganan penjadwalan fisioterapi. Penjadwalan fisioterapi masih menggunakan informasi konsultasi dengan aplikasi WhatsApp yang dilakukan secara manual oleh fisioterapis, dan kurangnya kemampuan untuk mengatur penjadwalan fisioterapis pada pasien. Pasien Bell's palsy pada Klinik Griya Fisioterapi rata-rata hanya menerima lima pasien dalam sehari, baik penyakit Bell's palsy dalam kategori ringan, sedang dan berat harus melakukan fisioterapis NMT selama 6 hari berturut-turut dan kemudian baru dapat dilakukan evaluasi. Hal itu dilakukan dengan tahapan melakukan konsultasi awal dengan pakar atau Fisioterapis dengan ketentuan gejala penyakit Bell's palsy, kemudian dilakukan pengecekan langsung nilai otot pasien, dan terakhir memberikan diagnosis sementara status penyakit pasien dengan kategori, ringan, sedang dan berat. Setiap penjadwalan dan evaluasi menjadi informasi yang penting bagi rekam medik untuk penanganan tahapan selanjutnya, sementara pada Klinik Griya Fisioterapi belum tersimpan dengan baik atau mengandalkan ingatan Fisioterapis saja. Sehingga diperlukannya sistem informasi dalam membantu penjadwalan fisioterapi sekaligus pencatatan rekam medis pasien, agar dapat terkontrol dan terkordinir dengan baik.

Metode pengembangan dalam sistem informasi yang dirancang menggunakan metode RUP (*Relational Unified Process*). RUP menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak, dimana RUP melakukan pendekatan perangkat lunak secara berulang-ulang (*iterative*), fokus pada arsitektur, lebih diarahkan berdasarkan pengerjaan perangkat lunak mulai dari pengumpulan awal dan penentuan kebutuhan sampai proses pembuatan program. Penggunaan *web* juga dilakukan berdasarkan kebutuhan sistem yang memudahkan dalam pengolahan data serta mempermudah tanpa harus dialakukannya instalasi sistem. Sehingga cocok untuk perancangan sistem informasi yang akan dibuat dalam penelitian yang diambil [5].

Berdasarkan uraian diatas penulis mengajukan sebuah penelitian dengan judul "Rancang Bangun Sistem Informasi Dan Konsultasi Penjadwalan Fisioterapi Penyakit Bell's Palsy (Studi Kasus Klinik Griya Fisioterapis)", sistem informasi ini



diharapkan dapat memberikan informasi konsultasi jadwal fisioterapi sehingga dapat mengatasi permasalahan awal penanganan penyakit Bell's palsy.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dapat dirumuskan pokok permasalahan berikut:

1. Bagaimana cara membantu memudahkan konsultasi awal dalam mendiagnosis penyakit Bell's palsy?
2. Bagaimana membangun suatu sistem informasi untuk membantu penjadwalan dan rekam medis fisioterapi pada penyakit Bell's palsy?

## **1.3 Batasan Masalah**

Dari permasalahan yang telah diuraikan di atas, terdapat batasan-batasan masalah dalam pembangunan sistem ini yaitu:

1. Sistem informasi dan konsultasi penjadwalan penyakit Bell's palsy merupakan sistem yang berbasis *web*.
2. Sistem yang dibuat berdasarkan pada pengetahuan pakar fisioterapi di Klinik Griya Fisioterapis.
3. Metode yang digunakan dalam pengembangan sistem yaitu metode *Relational Unified Process*.

## **1.4 Tujuan**

Adapun tujuan yang akan dicapai dari penelitian ini yaitu:

1. Memudahkan pakar untuk diagnosis awal penyakit Bell's palsy menggunakan fitur konsultasi berdasarkan data pakar fisioterapi di Klinik Griya Fisioterapis.
2. Membangun sistem informasi dan konsultasi penjadwalan untuk penyakit Bell's palsy dengan menggunakan metode pengembangan sistem *Relational Unified Process* berbasis *web*.

## **1.5 Manfaat**

Manfaat yang dapat diperoleh dari penelitian ini yaitu sebagai berikut:

- a. Mengetahui tingkatan penyakit Bell's palsy serta gejala-gejala yang menyebabkan gangguan tersebut.
- b. Menambah wawasan mengenai sistem informasi dan konsultasi penjadwalan penyakit Bells's palsy.

- c. Memberikan informasi mengenai jenis-jenis gangguan penyakit Bell's palsy melalui diagnosa gejala-gejala yang dialami oleh penderita.

## **1.6 Sistematika Penulisan**

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

### **1. BAB I PENDAHULUAN**

Bab ini menjelaskan dasar-dasar dari penyusunan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat serta sistematika penulisan laporan tugas akhir.

### **2. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI**

Bab ini menjelaskan tentang penelitian-penelitian terkait yang sudah dilakukan sebelumnya. Serta membahas teori-teori yang berhubungan dengan topik penelitian.

### **3. BAB III METODE PERANCANGAN**

Bab ini menguraikan tentang metode penelitian yang digunakan dalam merancang sistem. Seperti rencana pelaksanaan, alat, bahan, jalannya perancangan dengan metode yang telah ditentukan, perhitungan untuk hasil yang diharapkan serta cara pengujian sistem.

### **4. BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini merupakan pembahasan tentang analisis dan pembahasan sistem yang sudah dibuat, meliputi tampilan *database* dan implementasi sistem, serta hasil dari pengujian-pengujian yang telah dilakukan untuk mendapatkan hasil yang sesuai.

### **5. BAB V KESIMPULAN DAN SARAN**

Bab ini membahas kesimpulan dan saran dari hasil penelitian yang telah dilakukan.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Beberapa penelitian yang terkait dengan rancang bangun sistem informasi dan konsultasi penjadwalan fisioterapi penyakit Bell's palsy sebagai berikut.

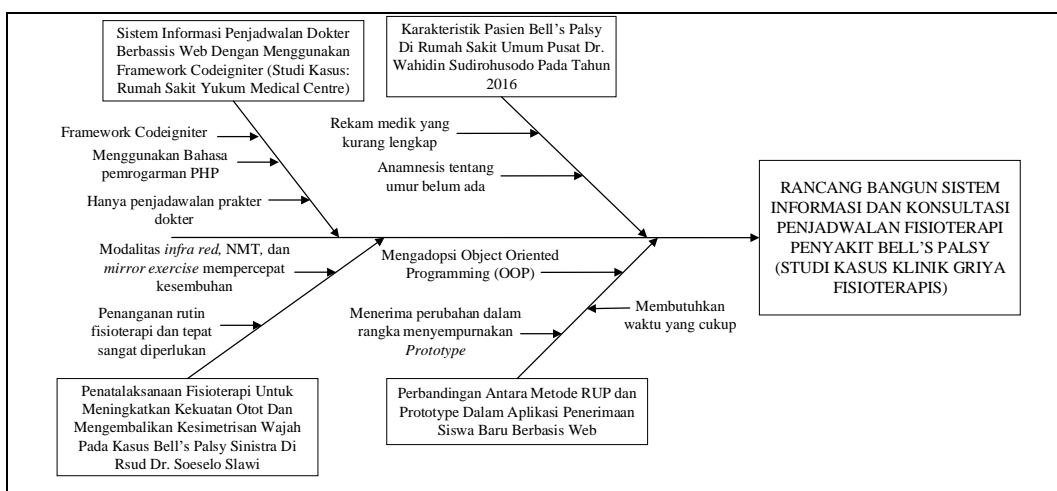
Hasil penelitian tentang "Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)" sistem informasi penjadwalan dokter ini diuji dengan menggunakan 3 metode pengujian yaitu *whitebox testing*, *software tester* dan *blackbox* (ISO 9126) dengan melakukan pengujian kuisioner kepada bagian informasi, dokter dan pasien. Hasil dari perhitungan kualitas sistem dihasilkan dengan presentase tanggapan sebesar 87,87 % dengan kriteria perhitungan sistem sangat baik [6].

Penelitian tentang "Perbandingan Antara Metode RUP dan Prototype Dalam Aplikasi Penerimaan Siswa Baru Berbasis Web" bertujuan untuk mengetahui gambaran secara sistematis, faktual dan akurat mengenai fakta-fakta dan sifat-sifat pada suatu objek penelitian tertentu pada pengembangan perangkat lunak. Metode pengembangan perangkat lunak model RUP dirancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan *prototype* yang sudah ada, sehingga pada akhirnya dapat menghasilkan perangkat lunak yang dapat diterima dan perubahan yang terjadi dapat dianggap bagian dari proses pengembangan itu sendiri [5].

Penelitian tentang "Karakteristik Pasien Bell's Palsy Di Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo Pada Tahun 2016" bertujuan untuk menganalisis karakteristik pasien yang terkena penyakit Bell's palsy dengan subjek terbanyak yang menderita Bell's palsy berada pada kelompok 46 - 55 tahun yaitu sebanyak 11 orang (35,5 %) dan terendah pada kelompok umur 6 - 10 tahun dan >65 tahun yaitu 0. Dari penelitian ini, didapatkan bahwa subjek terbanyak yang menderita Bell's palsy berada pada kelompok perempuan yaitu sebanyak 19 orang (61,3 %) dan terendah pada kelompok laki-laki sebanyak 12 orang ( 38,7 %) [1].

Penelitian tentang “Penatalaksanaan Fisioterapi Untuk Meningkatkan Kekuatan Otot Dan Mengembalikan Kesimetrisan Wajah Pada Kasus Bell’s Palsy Sinistra Di Rsud Dr. Soeselo Slawi” yang bertujuan untuk mempercepat penyembuhan pasien penderita Bell’s palsy dengan melakukan terapi secara cepat dan tepat, dimana dilakukan dengan menggunakan modalitas *Infra red* (IR), *massage*, *Neuro Muscular Taping* (NMT), dan *mirror exercise* serta edukasi untuk kondisi Bell’s palsy yang dilaksanakan di RSUD dr.Soeselo Slawi dan dilakukan sebanyak 3 kali terapi dengan nama pasien Tn. D umur 63 tahun diperoleh hasil meningkatnya kekuatan otot-otot wajah, meningkatnya kemampuan fungsional otot-otot wajah seperti saat istirahat, mengerutkan dahi memejamkan mata, serta membantu mempercepat proses kesembuhan pasien pada kondisi bell’s palsy [4].

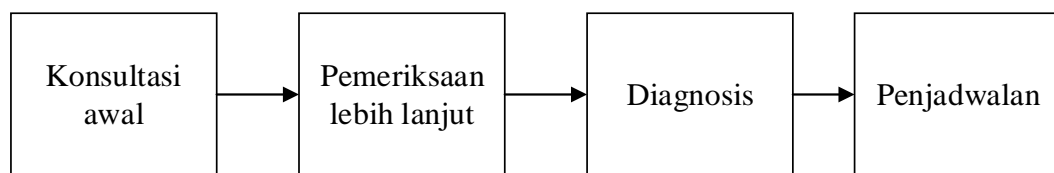
Berdasarkan penelitian yang pernah dilakukan, penulis bermaksud untuk menggunakan metode pengembangan perangkat lunak model *Relational Unified Process* (RUP) sebagai pengembangan perangkat lunak yang akan dibangun, dengan studi kasus yang diambil yaitu Klinik Griya Fisioterapis yang bertujuan untuk menghasilkan sistem informasi konsultasi penjadwalan fisioterapis yang sistematis berdasarkan konsultasi awal dan pemeriksaan pakar terkait nilai otot seperti pada Gambar 2.1 yang dijelaskan dalam *fishbone* diagram berdasarkan tinjauan pustaka.



Gambar 2.1 *Fishbone* Diagram berdasarkan tinjauan pustaka.

Adapun perancangan alur sistem sudah disesuaikan dengan *Standard Operating Procedure* dari pihak klinik, dimana dilakukan tahapan konsultasi secara

langsung antara pasien dengan Fisioterapis dengan gejala yang dialami, kemudian dilakukan pemeriksaan lebih lanjut dengan melakukan pemeriksaan nilai otot, selanjutnya dari hasil pemeriksaan dilakukan diagnosis bahwa pasien mengalami penyakit Bell's palsy dengan tingkatan ringan, sedang, berat sehingga harus dilakukan penjadwalan fisioterapi, hal tersebut digambarkan seperti pada Gambar 2.2.



Gambar 2.2 Alur sistem bekerja.

## 2.2 Dasar Teori

Berikut merupakan dasar teori yang berkaitan dengan rancang bangun sistem informasi dan konsultasi penjadwalan fisioterapi penyakit Bell's palsy (studi kasus klinik griya fisioterapis) sebagai berikut:

### 2.2.1 Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai suatu tujuan. Selain itu, sistem juga dapat didefinisikan sebagai sekumpulan objek-objek yang saling berelasi dan berinteraksi, serta hubungan antar objek bias dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan yang telah ditetapkan [7].

Sistem memiliki tiga komponen atau fungsi dasar yang berinteraksi, antara lain:

#### 1. *Input* (masukan)

Melibatkan penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses. Input yang dimaksud dalam hal ini berupa keseluruhan penginputan data yang berkaitan dengan transaksi dalam siklus pendapatan dan pengeluaran yang dilakukan oleh pihak yang berwenang.

#### 2. Proses

Melibatkan tahap transformasi yang mengubah *input* menjadi *output*. Yang dimaksud tahap disini mencakup penghitungan dan kalkulasi dari data-data transaksi siklus pendapatan dan pengeluaran yang masuk ke sistem.

### 3. *Output* (keluaran)

Melibatkan perpindahan elemen yang telah diproduksi oleh proses. *Output* yang dimaksud adalah laporan keuangan dan laporan produk yang berhasil dijual yang dihasilkan dari sistem informasi akuntansi *revenue cycle*. Dari berbagai definisi yang dikemukakan di atas dapat disimpulkan bahwa sistem adalah golongan dari komponen dan elemen yang disatukan untuk menggapai tujuan tertentu [8].

#### **2.2.2 Karakteristik Sistem**

Sebuah sistem mempunyai beberapa karakteristik, yang diantaranya yaitu sebagai berikut:

##### 1. Komponen atau Elemen (*Components*)

Suatu sistem terdiri dari komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.

##### 2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Adanya batas sistem, maka sistem dapat membentuk suatu kesatuan, karena dengan batas sistem ini, fungsi dan tugas dari subsistem satu dengan yang lainnya berbeda tetapi tetap saling berinteraksi.

##### 3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah segala sesuatu diluar batas sistem yang mempengaruhi operasi suatu sistem.

##### 4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan suatu media (penghubung) antara satu subsistem dengan subsistem lainnya yang membentuk satu kesatuan.

##### 5. Masukan (*Input*)

Input adalah energi atau sesuatu yang dimasukkan ke dalam suatu sistem yang dapat berupa masukan yaitu energi yang dimasukkan supaya sistem dapat beroperasi.

##### 6. Luaran (*Output*)

Merupakan hasil dari energi yang diolah dan diklarifikasi menjadi luaran yang berguna, juga menjadi luaran atau tujuan akhir sistem.

#### 7. Pengolah (*Process*)

Suatu sistem mempunyai bagian pengolah yang akan mengubah input menjadi output.

#### 8. Sasaran (*Objective*)

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

### **2.2.3 Sistem Informasi**

Sistem Informasi adalah kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna [9]. Yang terpenting dari sistem informasi adalah bagaimana teknologi disesuaikan dan digunakan untuk memungkinkan realisasi yang memenuhi berbagai aktor seperti individu, kelompok atau organisasi kebutuhan dan persyaratan informasi sehubungan dengan tujuan dan praktik tertentu [10]. Selain itu Sistem informasi adalah suatu system di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [11].

Pengertian lain juga mengatakan bahwa sistem informasi adalah suatu system di dalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian yang mendukung fungsi organisasi yang manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan pihak luar tertentu dengan laporan-laporan yang di perlukan [12].

Dari ketiga definisi yang dikemukakan oleh para ahli di atas dapat disimpulkan bahwa sistem informasi adalah suatu data yang dihimpun, dikategorikan, dan diproses sampai menjadi satu kesatuan informasi yang saling berkesinambungan satu sama lain dan saling mendukung sampai menjadi sebuah informasi yang bermanfaat bagi si penerimanya.

#### **2.2.4 Website**

*Website* merupakan sebuah kumpulan halaman-halaman *web* beserta *file-file* pendukungnya, seperti *file* gambar, video, dan *file* digital lainnya yang disimpan pada sebuah *web server* yang umumnya dapat diakses melalui internet. Atau dengan kata lain, *website* adalah sekumpulan folder dan *file* yang mengandung banyak perintah dan fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data, dsb. *Website* bisa digolongkan ke dalam beberapa jenis, seperti *website* pribadi, *website* komersial, *website* pemerintahan, dan *website* lembaga nirlaba [13].

#### **2.2.5 Framework Codeigniter**

*Framework codeigniter* atau juga sering disebut dengan CI adalah suatu framework MVC PHP yang sangat populer di dunia *web programmer*. *Framework CodeIgniter* diciptakan oleh EllisLab Incorporation, dimana Rick Ellis yang menjadi pendiri dan CEO EllisLab, perusahaan EllisLab ialah perusahaan *software development* yang menggunakan bahasa *scripting* PHP sebagai basisnya. EllisLab berada di Bend, Oregon negara bagian Deschutes, Amerika Serikat. *Codeigniter* pertama kali dirilis 28 Februari 2006 dan yang pertama kali keluar dengan versi beta 1.0 versi *codeigniter* sampai saat ini sudah sampai 3.4 [14].

#### **2.2.6 Bootstrap**

*Bootstrap* merupakan *framework* untuk membangun desain web secara responsif. Artinya, tampilan web yang dibuat oleh *bootstrap* akan menyesuaikan ukuran *layer* dan *browser* yang kita gunakan baik di *desktop*, tablet ataupun *mobile device*. Dengan *bootstrap* kita juga bisa membangun web dinamis ataupun statis [15].

#### **2.2.7 MySQL**

MySQL adalah Sebuah program *database* server yang mampu menerima dan mengirimkan datanya sangat cepat, multi *user* serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL Free Software* yang berada dibawah Lisensi GNU/GPL (*General Public License*).



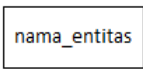
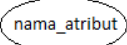
MySQL Merupakan sebuah *database* server yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Selain *database* server, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan sebagai *Client* maupun *server*.


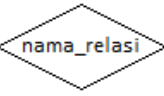

*Database* MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*) [16].

### 2.2.8 Entity Relationship Diagram

ERD (*Entity Relation Diagram*) merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data. ERD (*Entity Relation Diagram*) berguna untuk menggambarkan gambaran dari dunia nyata yang akan diterapkan pada suatu *database* sebuah sistem. ERD melihat objek nyata dapat sebagai sebuah entitas-entitas yang memiliki relasi antara entitas yang satu ataupun yang lain. Dengan ERD sendiri dapat membantu mengurangi kesalahan-kesalahan dalam melakukan perancangan *database* dari gambaran dunia nyata dan struktur *database* seperti redundansi data, hubungan-hubungan antara entitas, dan lain sebagainya [17]. Pada ERD terdapat beberapa simbol dengan cara penulisan di antaranya sebagai berikut:

Tabel 2.1 Simbol ERD.

Simbol	Nama	Keterangan
	Entitas / <i>Entitiy</i>	Merupakan data inti yang akan disimpan; bakal tabel pada basis data; biasanya mengacu pada benda yang terlibat dalam aplikasi yang akan dibuat.
	Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam satu entitas.

	Atribut kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas sebagai kunci akses <i>record</i> yang diinginkan.
	Relasi	Penghubung antar entitas; biasanya diawali dengan kata kerja.
	Asosiasi / <i>Association</i>	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.

## 2.2.9 UML

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *Class* dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.


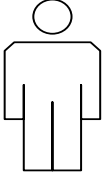

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan [18].

### 2.2.9.1 Use Case Diagram

*Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [18].

Tabel 2.2 Simbol use case diagram.

Simbol	Nama	Keterangan
	<i>Use Case</i>	Use Case mendeskripsikan urutan dari aksi yang memberikan nilai yang terukur dari aktor dan digambarkan sebagai lingkaran lonjong.
	<i>Actor</i>	Actor merupakan orang, organisasi, ataupun sistem eksternal yang melakukan role pada satu atau lebih interaksi dalam sistem. Aktor digambar dalam bentuk orang-orangan.
	Association / Asosiasi	Asosiasi merupakan penghubung antara aktor dan <i>use case</i> yang terlihat pada diagram <i>use case</i> dengan bentuk sebagai garis solid. Asosiasi terbentuk jika aktor ikut serta dengan interaksi yang digambarkan lewat <i>use case</i> . Asosiasi dimodelkan sebagai garis yang menghubungkan <i>use case</i> dan aktor antar satu dengan lainnya, dengan kepala panah pada ujung sisinya yang opsional untuk diberikan. Kepala panah tersebut sering digunakan untuk menentukan arah dari pelaku yang memicu pada hubungan atau menentukan aktor utama dalam <i>use case</i> .

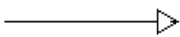
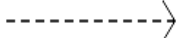

	<i>Generalization</i>	<i>Generalization</i> menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i> .
	<i>Include</i>	<i>Include</i> menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	<i>Extend</i>	<i>Extend</i> menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

### 2.2.9.2 Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *Class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu Nama (dan *stereotype*), Atribut, dan Metoda [18].

Tabel 2.3 Simbol Class diagram.

<b>Simbol</b>	<b>Nama</b>	<b>Deskripsi</b>
	<i>Class</i>	Kelas pada struktur sistem.
	Asosiasi / <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah / <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
	Kebergantungan / <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
	Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )



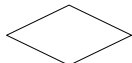
### 2.2.9.3 Activity Diagram

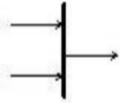
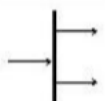

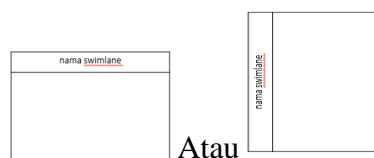
*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity* diagram merupakan state diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *Activity* diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas [18].

Tabel 2.4 Simbol Activity diagram.

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.


<p>Penggabungan / <i>Join</i></p> 	<p>Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p><i>Fork</i></p> 	<p>Menunjukkan adanya penggabungan aktivitas</p>
<p>Status Akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas harus memiliki status akhir.</p>
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

#### 2.2.9.4 Sequence Diagram

*Sequence* diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [18].

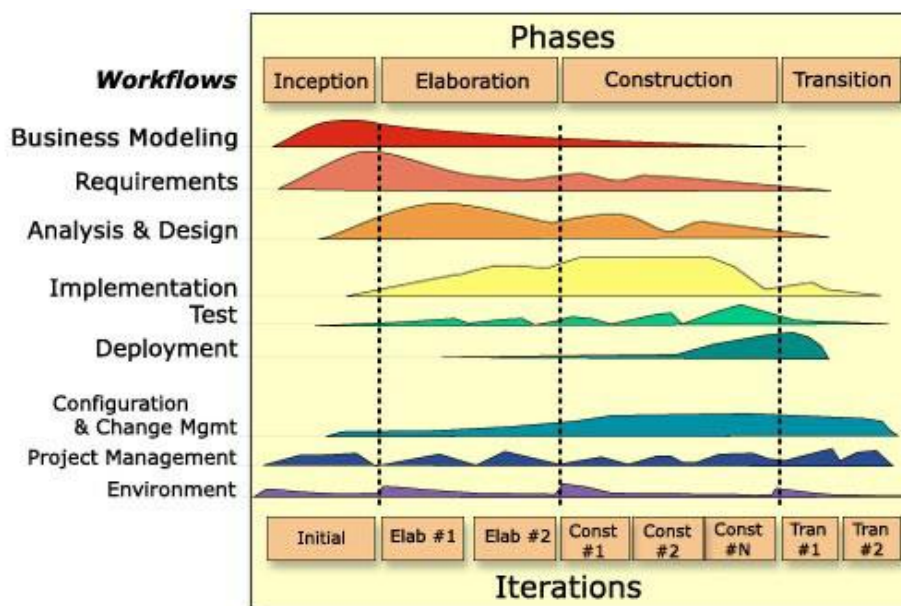
Tabel 2.5 Simbol Sequence diagram.

Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di bawah frase nama aktor.</p>

<i>lifeline</i>   	Menyatakan kehidupan suatu objek.
Objek  <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> Nama Objek:  Nama Kelas </div>	Menyatakan objek yang berinteraksi pesan.
Waktu aktif  <div style="border: 1px solid black; width: 20px; height: 20px; margin: 5px auto;"></div>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya

### 2.2.10 Relational Unified Process

*Relational Unified Process* (RUP) adalah pendekatan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang lebih baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakuisisi oleh IBM di bulan Febuaari 2003. Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada Gambar 2.3:



Gambar 2.3 Proses iterative RUP.

RUP memiliki 4 tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut adalah penjelasan untuk setiap fase RUP:

1. *Inception* (permulaan) tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).
2. *Elaboration* (perluasaan atau perencanaan) tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).
3. *Contruction* (kontruksi) tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahapan ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas atau tonggak kemampuan operasional awal.
4. *Transition* (transisi) tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas atau tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user* [5].

#### **2.2.11 Penyakit Bell's palsy**

Bell's palsy adalah suatu gangguan saraf fasialis perifer akut, yang biasanya mengenai hanya satu sisi wajah. Gambaran klinis bervariasi, tergantung lokasi lesi dari saraf fasialis sepanjang perjalanannya menuju otot. Gejala dan tanda yang dihasilkan tidak hanya pada serabut motorik termasuk ke otot stapedius, tetapi juga pada inervasi otonom kelenjar lakrimal, submandibular, sensasi sebagian telinga dan pengecapan pada dua pertiga lidah melalui korda timpani [1].



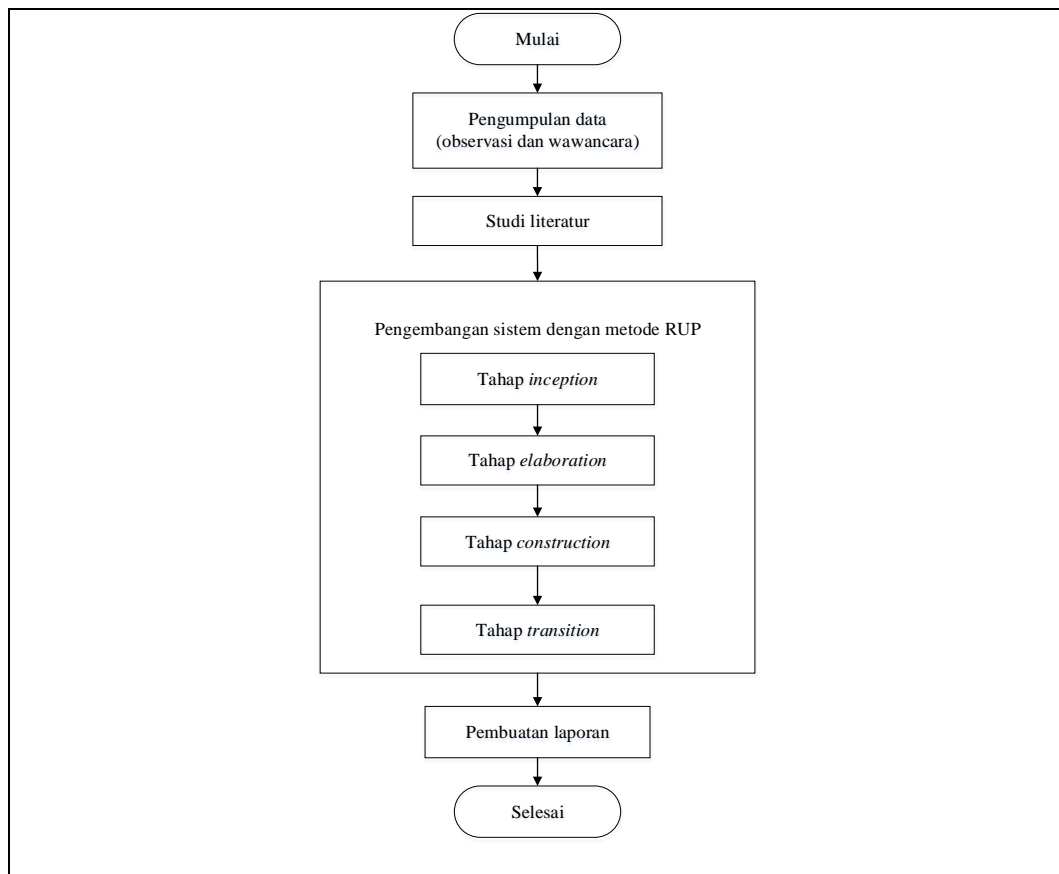
Pasien Bell's palsy biasanya datang dengan paralisis wajah unilateral yang terjadi secara tiba-tiba. Temuan klinis yang sering termasuk alis mata turun, dahi tidak berkerut, tidak mampu menutup mata, dan bila diusahakan tampak bola mata berputar ke atas (Bell's phenomen), sudut nasolabial tidak tampak dan mulut tertarik ke sisi yang sehat. Gejala lainnya adalah berkurangnya air mata, hiperakusis dan berkurangnya sensasi pengecapan pada dua pertiga depan [1].

Bell's Palsy (BP) menurut definisi sifatnya idiopatik. Semakin banyak bukti dalam literatur menunjukkan banyak hal kondisi dan patologi klinis potensial yang diketahui bermanifestasi, setidaknya sebagian, dengan periode wajah mengalami kelumpuhan [19]. Bell's palsy juga merupakan kelumpuhan pada saraf ketujuh, juga dikenal sebagai saraf wajah merupakan salah satu saraf kranial kedua belas (dua belas pasang saraf yang berasal dari otak, tidak seperti saraf tulang belakang yang berasal dari sumsum tulang belakang, yang dikenal dengan nama dan nomor mereka) [20].

## BAB III METODE PENELITIAN

### 3.1 Diagram Alir Pembuatan Sistem

Pembuatan sistem dengan menggunakan metode RUP dilakukan dengan pendekatan *iterative* (berulang-ulang), dimana dapat melakukan perulangan pada instruksi dan berhenti jika batas syarat sudah terpenuhi, dimana pada setiap tahapan dilakukan pengujian sistem sehingga mengakibatkan rendahnya kesalahan dan tingginya keberhasilan. Dalam pengembangan sistem menggunakan beberapa tahapan yang mengikuti metode RUP seperti yang terlihat pada Gambar 3.1 berikut.



Gambar 3.1 Tahapan pengembangan sistem [21].

Berikut merupakan penjelasan mengenai tahapan pengembangan sistem berdasarkan pada Gambar 3.1.

### 3.2 Pengumpulan data

Pengumpulan data pada tugas akhir ini dilakukan dengan dua cara yaitu observasi dan wawancara.

### **3.2.1 Observasi**

Pada proses observasi yang dilakukan, penulis mengamati secara langsung bagaimana proses pengelolaan data pasien penyakit Bell's palsy, proses melakukan penjadwalan penyakit Bell's palsy dan evaluasi pasien penyakit Bell's palsy. Hasil yang didapat dari observasi yang dilakukan yaitu sebuah form konsultasi awal dengan analisis status dari fisioterapis.

### **3.2.2 Wawancara**

Pada wawancara yang dilakukan penulis di Klinik Griya Fisioterapis dengan Bapak R. Arif Trianto C, S.ST.FT., S.Th.I., NMTCP-SM selaku fisioterpis, dijelaskan mengenai proses penjadwalan fisioterapi penyakit Bell's palsy yang hanya dilakukan menggunakan media sosial yaitu Whatsapp. Proses yang dilakukan dalam penjadwalan fisioterapi masih menggunakan aplikasi Whatsapp dengan tampungan lima pasien setiap harinya. Penanganan pasien Bell's palsy tidak dapat dilakukan dengan menggunakan sistem, diharuskan untuk melakukan konsultasi awal dengan pakar fisioterapis secara langsung, dengan mengetahui gejala yang di alami pasien Bell's palsy maka fisioterpis dapat menentukan status awal dari penderita Bell's palsy berdasarkan tingkat penyakit rendah, sedang, atau berat yang di derita pasien, serta dilakukan fisioterapi sebanyak 6 kali diawal scara berturut-turut untuk pasien awal penyakit Bell's palsy. Hal itu dilakukan sesuai anjuran fisioterpi dengan prosedur fisioterapis untuk menghindari dampak penyakit Bell's palsy yaitu cacat seumur hidup. Berdasarkan permasalahan diatas dapat dilakukan pembangunan sistem informasi sebagai sarana membantu dalam melakukan konsultasi penjadwalan fisioterapi penyakit Bell's palsy.

### **3.3 Studi literatur**

Pada penelitian ini, penulis melakukan studi literatur dengan cara mengumpulkan data dari berbagai penelitian, jurnal maupun sumber lain yang berkaitan dengan penelitian ini. Dilakukanya studi literatur bertujuan untuk melihat kekurangan serta kelebihan yang ada pada skripsi, jurnal maupun sumber lainnya yang nantinya digunakan sebagai acuan untuk mengembangkan sistem yang akan dibangun.

### 3.4 Pengembangan sistem dengan metode RUP

Pada pengembangan sistem dengan metode RUP terdapat 4 tahapan yaitu tahap *inception*, *elaboration*, *construction*, dan *transition*.

#### 3.4.1 Tahap *Inception*

Pada tahap ini yang dilakukan adalah membuat *business modeling*, *requirement* kebutuhan, analisis dan desain, serta *test inception*. Berikut ini merupakan penjelasan dari masing-masing proses pada tahap *inception*.

##### 3.4.1.1 *Business Modelling*

Bisnis model *canvas* yang digunakan pada tahap awal untuk memulai perancangan sistem seperti pada Gambar 3.2.

Bussines Model Canvas				
<b>Key Patners</b> -Fisioterapis -Pasien -Admin	<b>Key Activities</b> Memudahkan dalam melakukan konsultasi awal penyakit Bell's palsy, memudahkan mencatat rekam medik pasien, dan memudahkan pasien untuk memperoleh jadwal konsultasi	<b>Value Proposition</b> Sistem Informasi dan Konsultasi Penjadwalan penyakit Bell's Palsy	<b>Customer Relationship</b> -Customer service atau Admin	<b>Customer Segments</b> -Fisioterapis -Pasien
	<b>Key Resources</b> -Fisioterapis -Admin -Web developer		<b>Channels</b> -Website Klinik yang telah di hosting -user guide	
<b>Cost Structure</b> Hosting sistem		<b>Revenue Streams</b> Fisioterapis dipermudah dalam melakukan konsultasi awal penyakit Bell's palsy dan mempermudah dalam melakukan rekaman medik pasien. Serta pasien juga dimudahkan dalam memperoleh jadwal konsultasi		

Gambar 3.2 *Business modelling canvas* rancang bangun sistem informasi dan konsultasi penjadwalan fisioterapi penyakit bell's palsy (studi kasus klinik griya fisioterapis).

#### 1. *Customer Segments*

*Customer Segments* dalam penelitian yang akan dilakukan adalah mempermudah Fisioterpis Bell's palsy dalam melakukan penjadwalan dan pencatatan rekam medik, dimana pasien Bell's palsy dimudahkan memperoleh jadwal konsultasi.

#### 2. *Value Proposition*

*Value Proposition* dalam penelitian yang akan dilakukan merupakan sebuah *website* untuk Klinik Griya Fisioterapi dengan sistem informasi dan konsultasi penjadwalan penyakit Bell's palsy.

### 3. *Channels*

*Channels* dalam penelitian yang akan dilakukan merupakan sebuah sistem yang telah di *hosting* sesuai dengan fungsi yang telah ditentukan, seperti melakukan *input* data pasien, jadwal fisioterapi, dan proses menambah data rekam medik.

### 4. *Customer Relationship*

*Customer Relationship* merupakan *customer service* yang akan membantu pengguna jika mengalami kesulitan ataupun memiliki pertanyaan dengan mengirim *email* ke *customer service*.

### 5. *Revenue Streams*

*Revenue streams* dalam penelitian untuk mempermudah Fisioterapis dalam mengelola konsultasi dan penjadwalan serta rekam medik pasien. Digunakannya sistem informasi konsultasi penjadwal Bell's palsy studi kasus di Klinik Griya Fisioterapis, maka akan mempermudah dalam melakukan konsultasi dan penjadwalan sesuai dengan prosedur yang berlaku.

### 6. *Key Resource*

*Key Resource* dalam penelitian merupakan *web developer* untuk membangun sistem dan *server* untuk menyimpan data pengguna ataupun segala aktivitas yang dilakukan dalam sistem.

### 7. *Key Activities*

*Key Activities* yang akan dilakukan yaitu mempermudah Fisioterapis untuk melakukan konsultasi penjadwalan dan pencatatan rekam medik pasien Bell's palsy.

### 8. *Key Partners*

*Key Partners* dalam penelitian yang akan dilakukan meliputi Fisioterapis di Klinik Griya Fisioterapi, *admin*, dan pasien penyakit Bell's palsy sesuai kebutuhan *website* yang akan dibangun.

## 9. Cost Structure

*Cost Structure* dalam penelitian ketika diimplementasikan oleh pihak klinik akan yang dilakukan adalah biaya *hosting website* yang akan dibangun agar dapat bekerja sesuai dengan yang dibutuhkan. Berikut ini merupakan rincian dari biaya *hosting website* seperti pada Tabel 3.1.

Tabel 3.1 *Cost Structure*

No	Nama	Biaya
1.	<i>Server</i>	Rp. 350.000,00/tahun
2.	<i>Domain</i>	Rp. 125.000,00/tahun

### 3.4.1.2 Requirement

Kebutuhan sistem dibedakan menjadi kebutuhan fungsional dan non fungsional. Kebutuhan fungsional terdiri dari kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### a. Kebutuhan fungsional sistem

Kebutuhan Fungsional adalah kebutuhan yang berisi proses-proses apa saja/layanan apa saja yang nantinya harus disediakan oleh sistem, mencakup bagaimana sistem harus bereaksi pada *input* tertentu dan bagaimana perilaku sistem pada situasi tertentu. Pada pemaparan kebutuhan fungsional sistem terdapat 13 aksi yang akan dilakukan oleh 3 aktor yaitu Admin, Fisioterapis, dan Pasien. Masing-masing aktor memiliki hak akses yang berbeda seperti yang terlihat pada Tabel 3.2 berikut.

Tabel 3.2 Kebutuhan fungsional sistem.

No	Kebutuhan Fungsional	Aktor
1	Menambah data pasien	Admin
2	Mengubah data pasien	
3	Menghapus data pasien	
4	Mencetak rekam medik	
5	Melihat grafik pasien	
6	Menambah jadwal fisioterapi	
7	Mengubah jadwal fisioterapi	
8	Menghapus jadwal fisioterapi	

9	Melihat jadwal fisioterapi	
10	Melihat grafik pasien	Fisioterapis
11	Melihat rekam medik	
12	Melihat jadwal fisioterapi	
13	Menambah rekam medik	
14	Melihat jadwal	Pasien

**b. Kebutuhan non-fungsional sistem**

Kebutuhan non-fungsional merupakan kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsional juga sering disebut sebagai batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dan lain lain. Kebutuhan non-fungsional sistem terdiri dari 2 kebutuhan yaitu seperti yang terdapat pada Tabel 3.3 berikut.

Tabel 3.3 Kebutuhan non-fungsional sistem.

No	Kebutuhan Non-Fungsional
1	Tampilan disesuaikan dengan fungsi yang sudah dibuat.
2	Sistem dapat dijalankan di semua <i>web browser</i> seperti <i>Google Chrome</i> , <i>Mozilla Firefox</i> , <i>UC Browser</i> dan lain sebagainya.

**1) Kebutuhan perangkat keras (*Hardware*)**

Kebutuhan akan perangkat keras (*hardware*) untuk mengimplementasikan sistem yaitu seperti yang terdapat pada Tabel 3.4 berikut.

Tabel 3.4 Kebutuhan perangkat keras.

No	Keterangan	Spesifikasi	Fungsi
1	<i>Processor</i>	Intel Core i7	Untuk memproses data yang diterima dari <i>input</i> sehingga menghasilkan <i>output</i>
2	<i>Ram</i>	4 GB	Sebagai pendukung bagi <i>processor</i> dalam <i>computing</i>
3	<i>Hard disk</i>	1 TB	Sebagai media penyimpanan data dan program pada komputer yang bersifat permanen selama tidak rusak
4	<i>Monitor</i>	Led 14 Inchi	Untuk melihat tampilan program

## 2) Kebutuhan perangkat lunak (*Software*)

Kebutuhan akan perangkat lunak (*software*) pada pengembangan dan pengimplementasian sistem yaitu seperti yang terdapat pada Tabel 3.5 berikut.

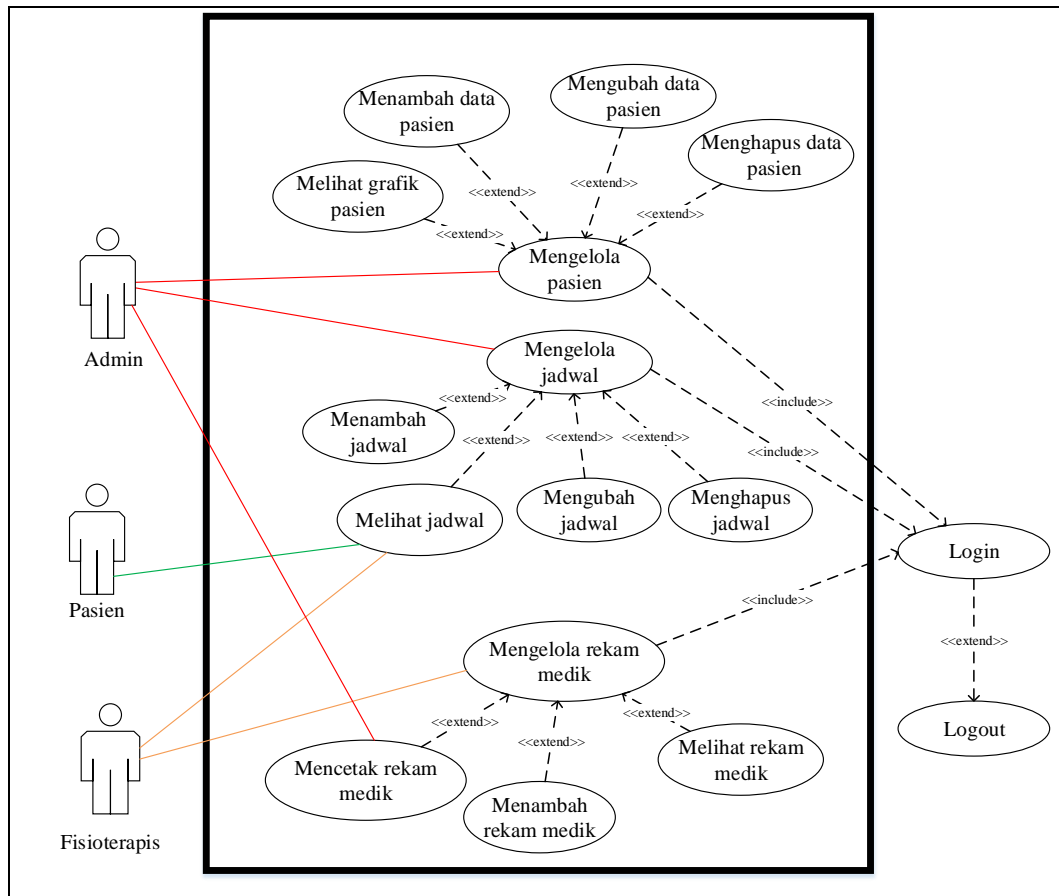
Tabel 3.5 Kebutuhan perangkat lunak.

No	Keterangan	Spesifikasi	Fungsi
1	Sistem operasi	Windows 10	Perangkat lunak sistem yang mengatur sumber daya dari perangkat keras dan perangkat lunak, serta sebagai jurik ( <i>daemon</i> ) untuk program komputer.
2	<i>Xampp</i>	Version 3.2.1	Sebagai <i>server</i> yang berdiri sendiri ( <i>localhost</i> ), yang terdiri dari beberapa program diantaranya <i>Apache</i> HTTP Server, <i>MySQL</i> database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan <i>Perl</i> .
3	<i>Visual Studio Code</i>	Version 1.43	<i>Text editor</i> yang akan digunakan untuk membuat <i>code</i> program dari aplikasi yang akan dibuat
4	<i>Google Chrome</i>	70.0.3538.110	Sebagai <i>browser</i> untuk menjalankan sistem



### 3.4.1.3 Analysis and Design inception

Analisis aplikasi pada tahap *inception* dijelaskan dengan menggunakan *use case diagram* seperti pada Gambar 3.3 berikut.



Gambar 3.3 Use case diagram sistem.

Gambar 3.3 merupakan *use case diagram* sistem setelah dilakukan perubahan. Berdasarkan Gambar 3.3 diketahui bahwa pada sistem ini terdapat 3 aktor yang berperan sebagai Admin, Fisioterapis dan Pasien, dimana masing-masing aktor memiliki akses yang berbeda-beda. Admin memiliki 9 akses yaitu menambah pasien, mengubah pasien, menghapus pasien, menambah jadwal, mengubah jadwal, menghapus jadwal, mencetak rekam medik, melihat jadwal, dan melihat grafik pasien. Kemudian untuk Fisioterapis memiliki akses 3 akses yaitu, melihat grafik pasien, melihat rekam medik, dan menambah rekam medik. Sedangkan untuk Pasien memiliki 1 akses yaitu dapat melihat jadwal. Semua akses yang diberikan dapat dilakukan dengan melakukan *login* ke sistem.

#### 3.4.1.4 Test Inception

Pada tahap ini, pengujian bertujuan untuk memvalidasi kebutuhan sistem dan *use case* yang telah dirancang. Pengujian melibatkan pakar fisioterapis Klinik Griya Fisioterapis dimana pengujian dilakukan dengan mencocokkan *use case* berdasarkan kebutuhan fungsional sistem. Proses validasi dimulai dengan peneliti memberikan tabel isian yang kemudian diisi dari penguji pada klinik. Hasil validasi dapat dilihat pada Tabel 3.6 berikut.

Tabel 3.6 Validasi kesesuaian *use case* dengan kebutuhan sistem.

No	Kebutuhan fungsional sistem	Use Case		
		Admin	Fisioterapis	Pasien
1	Mengelola pasien (menambah, mengubah, dan menghapus)	✓		
2	Mengelola jadwal (menambah, mengubah, dan menghapus)	✓		
3	Mencetak rekam medik	✓		
4	Melihat grafik pasien	✓	✓	
5	Melihat rekam medik		✓	
6	Menambah rekam medik		✓	
7	Melihat jadwal	✓	✓	✓

Berdasarkan validasi yang dilakukan diperoleh hasil bahwa kebutuhan dan *use case* sistem yang dirancang telah sesuai dengan yang diinginkan oleh pengguna. Maka proses dapat dilanjutkan menuju tahap selanjutnya yaitu tahap *elaboration*.

#### 3.4.2 Tahap Elaboration

Tahap *elaboration* merupakan kelanjutan dari tahap *inception* di mana pada tahapan *analysis and design* akan dilengkapi dengan menambahkan *activity diagram*, *sequence diagram*, *class diagram* dan desain *database*. Dilanjutkan dengan tahap *implementation* yaitu *prototype* yang berupa *digital prototype* dari desain *interface* sistem.

### 3.4.2.1 Requirements Elaboration

Terdapat penambahan kebutuhan fungsional sistem dari sebelumnya yaitu menu melihat grafik pasien dan mencetak rekam medik. Hal itu dilakukan oleh *user admin* sebagai penunjang kebutuhan klinik.

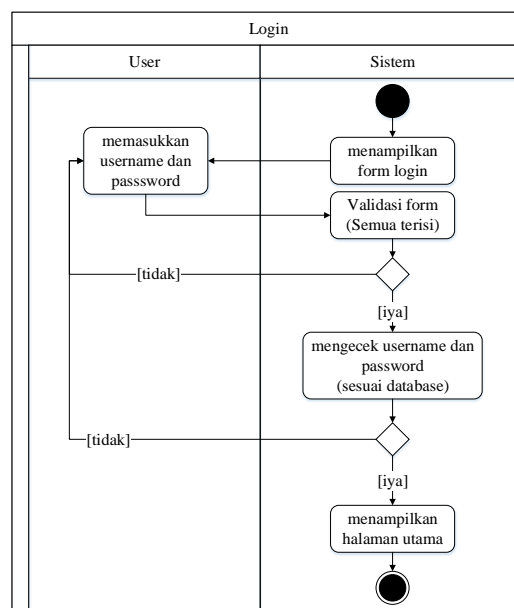
### 3.4.2.2 Analysis and Design

Analisis dan desain pada tahap *elaboration* yaitu perancangan *activity diagram*, *sequence diagram*, *class diagram* dan desain *database*.

#### 1. Activity Diagram

Berikut ini adalah *Activity Diagram* dari Rancang Bangun Sistem Informasi dan Konsultasi Penjadwalan Fisioterapi Penyakit Bell's Palsy (Studi Kasus Klinik Griya Fisoterapis):

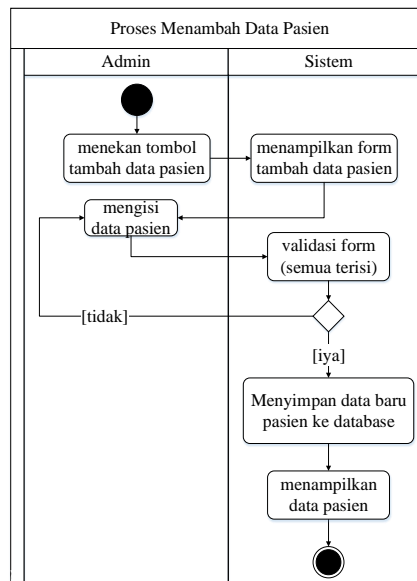
##### a) Proses Login



Gambar 3.4 Proses login.

Pada Gambar 3.4 merupakan proses *login* yang dilakukan oleh *user*. *User* dapat melakukan aktivitas *login* untuk masuk ke sistem dengan memasukkan *username* dan *password* yang telah terdaftar pada sistem, kemudian sistem akan melakukan validasi pengisian semua *form*. Jika masukan *user* benar atau sesuai dengan *database*, maka *user* akan diarahkan ke halaman utama (sesui dengan level) dan jika salah akan diarahkan melakukan pengisian ulang kembali.

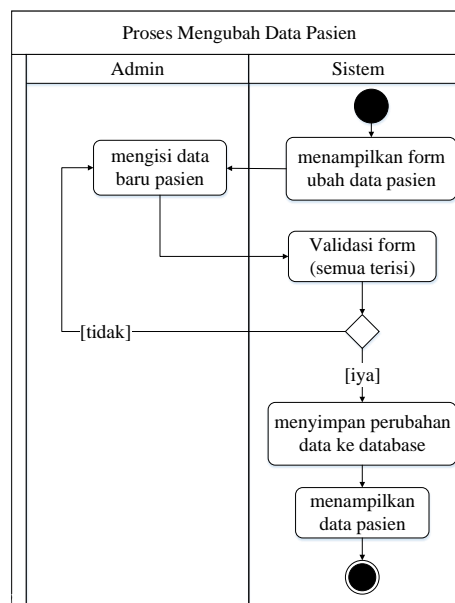
## b) Proses Menambah Data Pasien



Gambar 3.5 Proses menambah data pasien.

Pada Gambar 3.5 merupakan proses menambah data pasien yang dilakukan oleh *admin*. *Admin* harus mengisi *form* data pasien yang kemudian akan divalidasi dan disimpan ke dalam *database*.

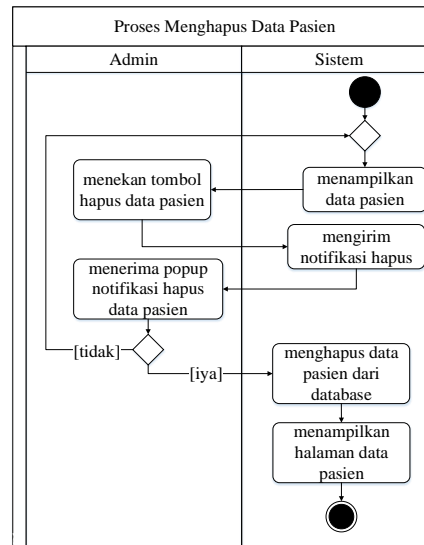
## c) Proses Mengubah Data Pasien



Gambar 3.6 Proses mengubah data pasien.

Pada Gambar 3.6 merupakan proses mengubah data pasien yang dilakukan oleh *admin*. *Admin* harus mengisi *form* data pasien yang kemudian akan divalidasi dan disimpan ke dalam *database*.

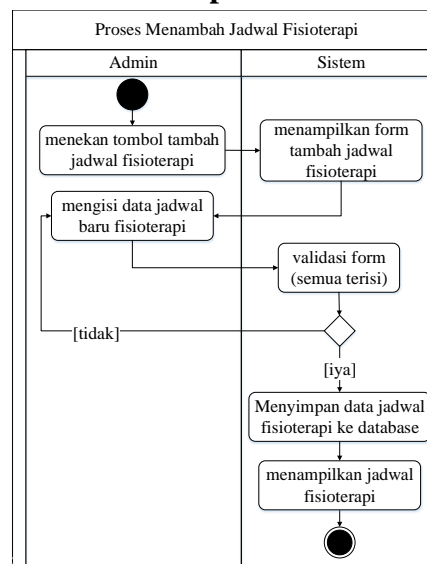
#### d) Proses Menghapus Data Pasien



Gambar 3.7 Proses menghapus data pasien.

Pada Gambar 3.7 merupakan proses menghapus data pasien yang dilakukan oleh *admin*. *Admin* dapat menghapus data pasien dengan menekan tombol hapus data pasien, kemudian akan direspon oleh sistem untuk menghapus data pasien sesuai dengan *database*.

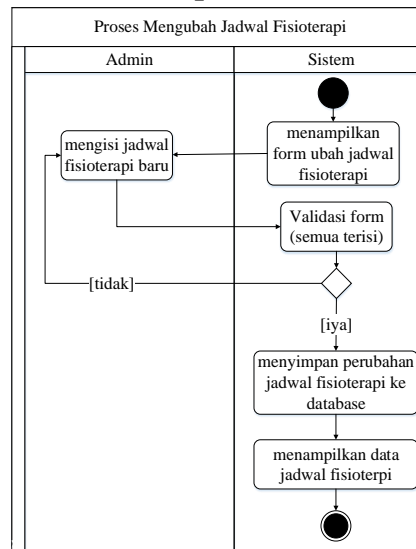
#### e) Proses Menambah Jadwal Fisioterapi



Gambar 3.8 Proses menambah jadwal fisioterapi.

Pada Gambar 3.8 merupakan proses menambah jadwal fisioterapi yang dilakukan oleh *admin*. *Admin* harus mengisi *form* data pasien yang kemudian akan divalidasi dan disimpan ke dalam *database*.

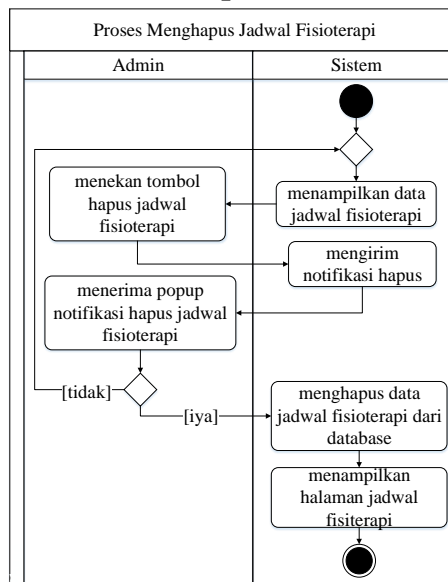
**f) Proses Mengubah Jadwal Fisioterapi**



Gambar 3.9 Proses mengubah jadwal fisioterapi.

Pada Gambar 3.9 merupakan proses mengubah jadwal fisioterapi yang dilakukan oleh *admin*. *Admin* harus mengisi *form* jadwal fisioterapi yang baru kemudian akan divalidasi dan disimpan ke dalam *database*.

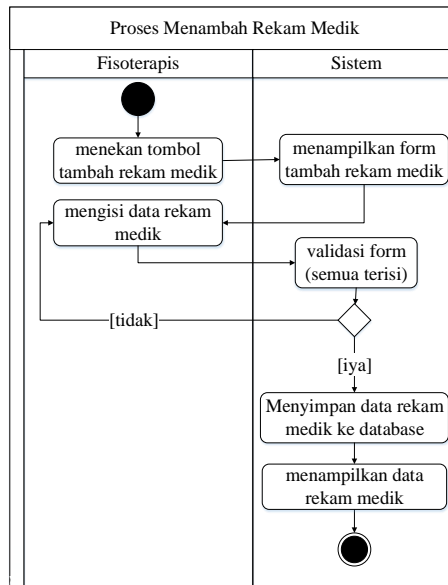
**g) Proses Menghapus Jadwal Fisioterapi**



Gambar 3.10 Proses menghapus jadwal fisioterapi.

Pada Gambar 3.10 merupakan proses menghapus data pasien yang dilakukan oleh *admin*. *Admin* dapat menghapus data pasien dengan menekan tombol hapus data pasien, kemudian akan direspon oleh sistem untuk menghapus data pasien sesuai dengan *database*.

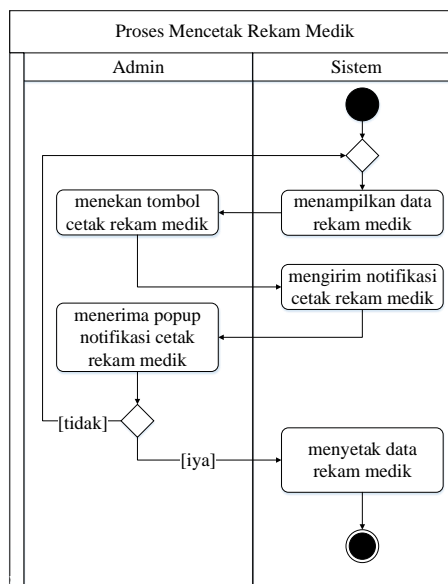
## h) Proses Menambah Rekam Medik



Gambar 3.11 Proses menambah rekam medik.

Pada Gambar 3.11 merupakan proses menambah rekam medik yang dilakukan oleh *fisioterapis*. *Fisioterapis* harus mengisi *form* data rekam medik yang kemudian akan divalidasi dan disimpan ke dalam *database*.

## i) Proses Mencetak Rekam Medik

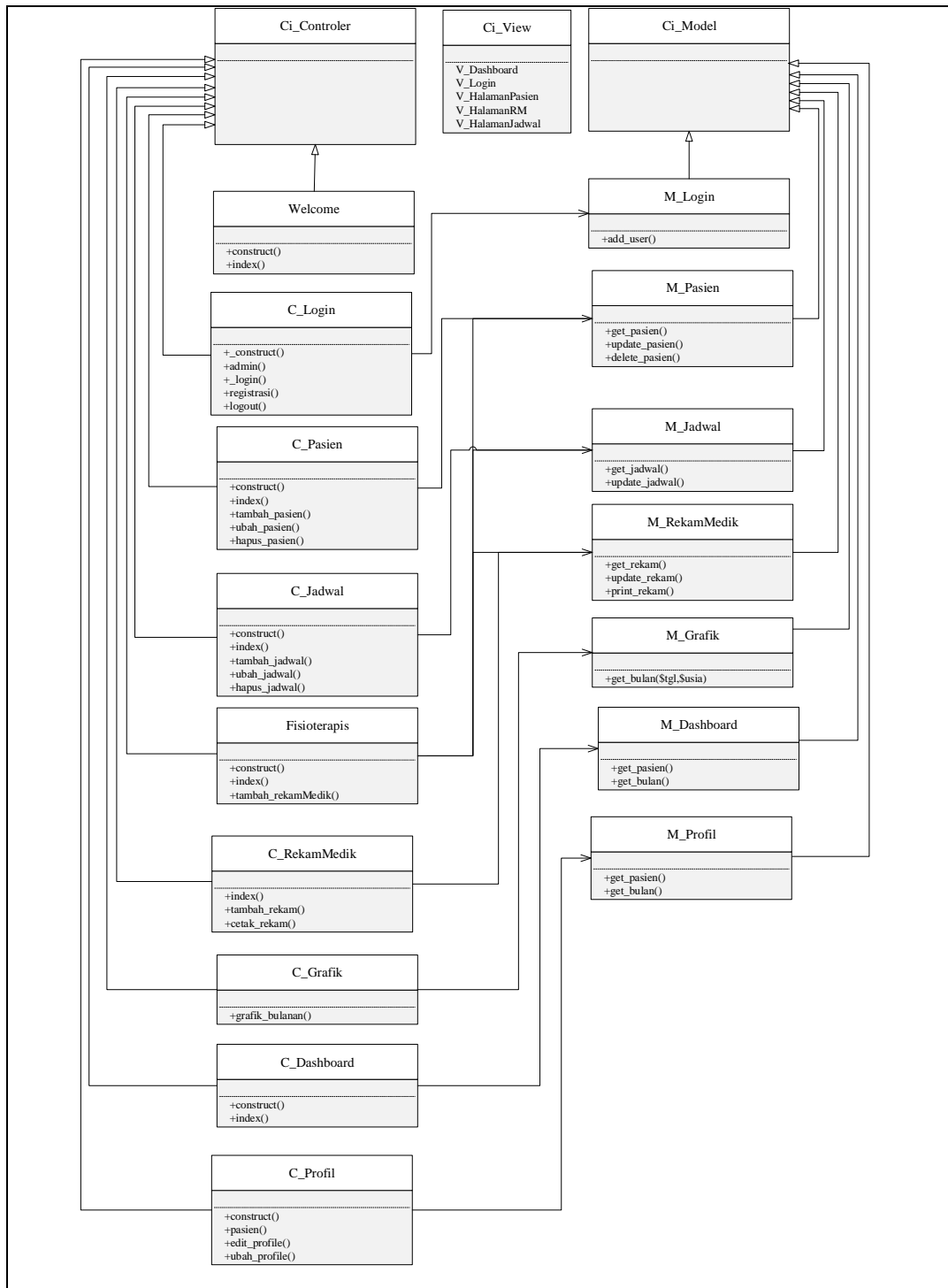


Gambar 3.12 Proses mencetak rakam medik.

Pada Gambar 3.12 merupakan proses mencetak rekam medik yang yang dilakukan oleh *admin*. *Admin* dapat melakukan cetak data rekam medik dengan menekan tombol cetak, maka data rekam medik akan langsung tercetak.

## 2. Class Diagram

Keterhubungan antar kelas yang terdapat pada sistem dapat dilihat pada Gambar 3.13 berikut ini:



Gambar 3.13 Class diagram.

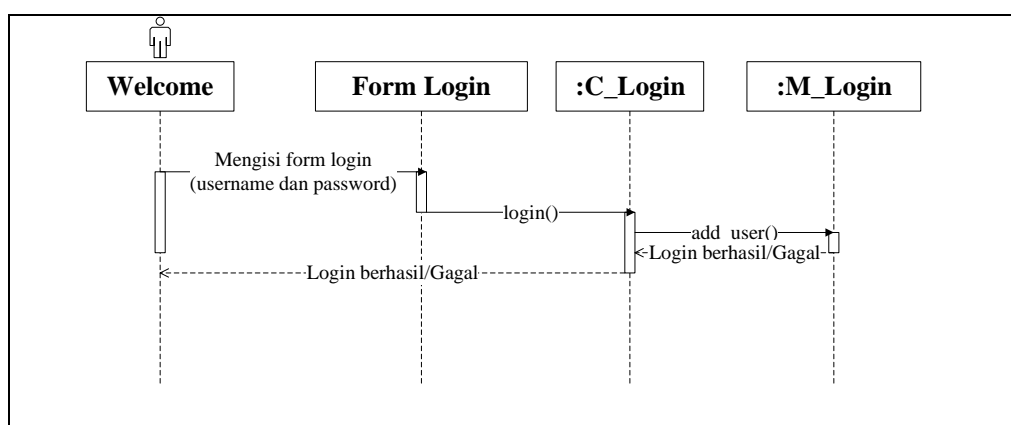


Berdasarkan Gambar 3.13 dapat dilihat bahwa sistem dibangun menggunakan metode analisa berorientasi objek dan menggunakan pola arsitektur MVC (*Modeling Views Controllers*), sehingga dilakukan perancangan kelas-kelas yang saling terhubung terdiri dari *Models*, *Views*, dan *Controllers*. Pada *Controllers* terdiri dari Welcome, C\_Login, Fisioterapis, C\_Pasien, C\_Jadwal, C\_RekamMedik, C\_Grafik, C\_Dashboard, dan C\_Profil. Pada *Views* terdapat pada *form* yang akan ditampilkan sistem, sedangkan untuk *Models* terdiri dari Login\_Model, M\_Pasien, M\_Jadwal, M\_RekamMedik, dan M\_Grafik. Pada *Controller* Login berhubungan dengan Fisoterapis, C\_Pasien, dan M\_Login. *Controller* Fisioterapis berhubungan dengan M\_Pasien, M\_Jadwal dan M\_RekamMedik. *Controller* C\_Pasien berhubungan dengan M\_Pasien. *Controller* C\_Jadwal berhubungan dengan M\_Jadwal. *Controller* C\_RekamMedik berhubungan dengan M\_RekamMedik. *Controller* Grafik berhubungan dengan M\_Grafik. *Controller* C\_Dashboard berhubungan dengan M\_Dashboard. *Controller* C\_Profil berhubungan dengan M\_Profil.

### 3. Sequence Diagram

Berikut ini merupakan *Sequence Diagram* pada Rancang Bangun Sistem Informasi Dan Konsultasi Penjadwalan Fisioterapi Penyakit Bell's Palsy (Studi Kasus Klinik Griya Fisioterapis) sebagai berikut:

#### 1. Proses Login

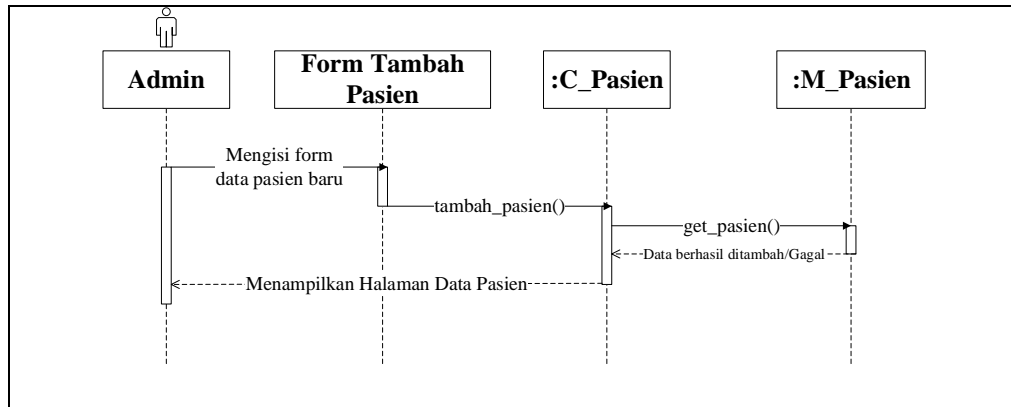


Gambar 3.14 Proses Login Admin, Fisioterapis dan Pasien.

Pada Gambar 3.14 merupakan *sequence diagram* proses login, dimulai ketika *user* mengisi *username* dan *password* pada halaman *form login*.

Kemudian memanggil fungsi `login()` pada *controller* `C_Login` untuk mengecek hasil data yang di *input*. Selanjutnya memanggil fungsi `add_user()` pada model `M_Login`. Dari model mengirimkan nilai login berhasil atau gagal apabila berhasil controller menampilkan halaman *dashboard*.

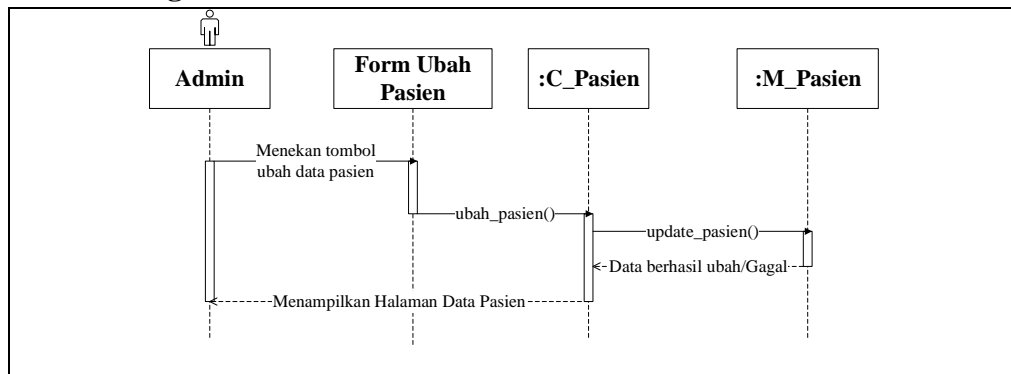
## 2. Proses Menambah Pasien



Gambar 3.15 Proses menambah pasien.

Pada Gambar 3.15 merupakan proses menambah pasien, dimulai ketika *Admin* mengisi data pasien baru pada halaman *form* tambah pasien. Kemudian hasil *input* data akan dikirimkan ke kelas `C_Pasien` menggunakan fungsi `tambah_pasien()`. Selanjutnya data tersebut akan ditambahkan ke dalam *database* menggunakan fungsi `get_pasien()` pada kelas `M_Pasien`. Jika berhasil ditambahkan, maka *Admin* diarahkan menuju halaman data pasien.

## 3. Proses Mengubah Data Pasien

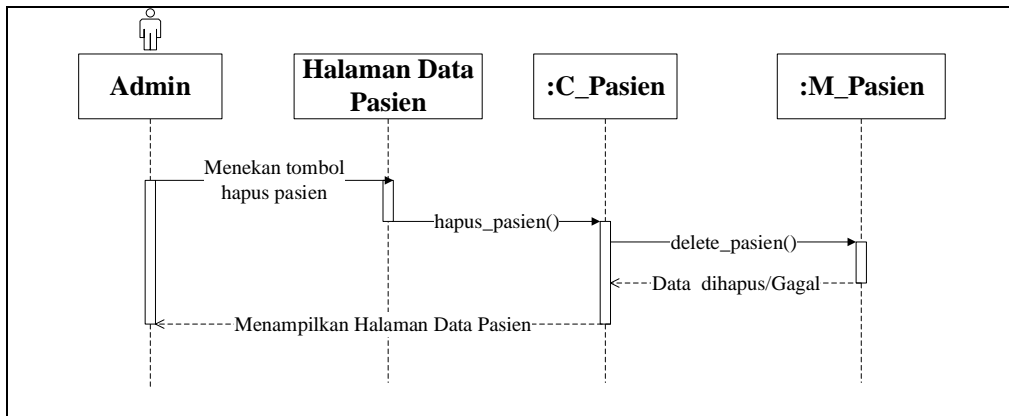


Gambar 3.16 Proses mengubah data pasien.

Pada Gambar 3.16 merupakan proses mengubah data pasien, dimulai ketika *Admin* menekan tombol ubah data pasien pada halaman *form* ubah pasien. Kemudian hasil *input* data akan dikirimkan ke kelas `C_Pasien`

menggunakan fungsi `ubah_pasien()`. Selanjutnya data tersebut akan dikonfirmasi ke dalam *database* menggunakan fungsi `update_pasien()` pada kelas `M_Pasien`. Jika berhasil ditambahkan, maka *Admin* diarahkan menuju halaman data pasien.

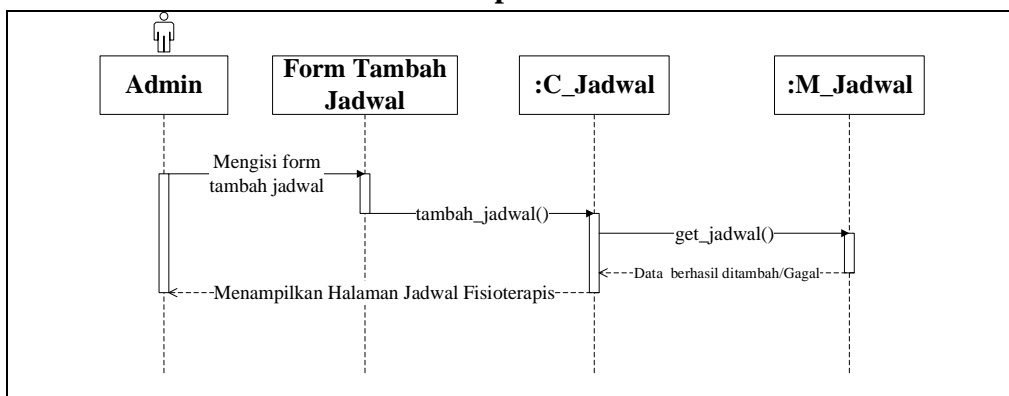
#### 4. Proses Menghapus Data Pasien



Gambar 3.17 Proses menghapus data pasien.

Pada Gambar 3.17 merupakan proses menghapus data pasien, dimulai ketika *Admin* menekan tombol hapus pasien pada halaman data pasien. Kemudian dilakukan konfirmasi penghapusan data pasien pada kelas `C_Pasien` menggunakan fungsi `hapus_pasien()`. Selanjutnya data tersebut akan dihapus dari dalam *database* menggunakan fungsi `delete_pasien()` pada kelas `M_Pasien`. Jika data pasien berhasil dihapus, maka *Admin* diarahkan menuju halaman data pasien.

#### 5. Proses Menambah Jadwal Fisioterapi

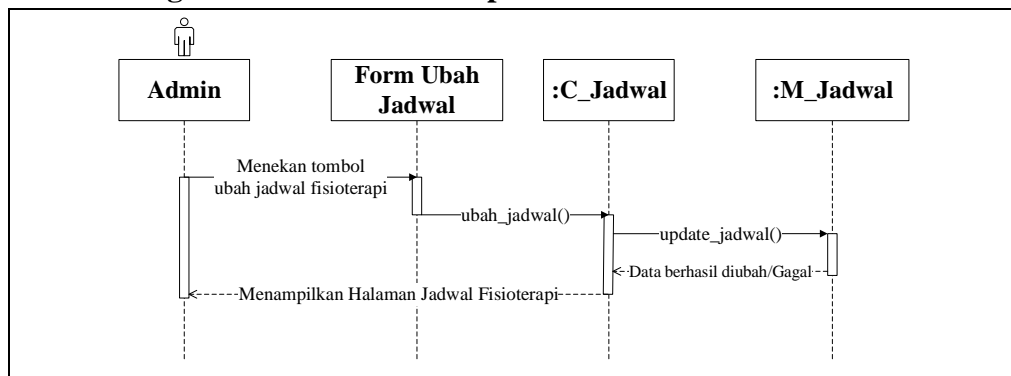


Gambar 3.18 Proses menambah jadwal fisioterapi.

Pada Gambar 3.18 merupakan proses menambah jadwal fisioterapis, dimulai ketika *Admin* mengisi jadwal baru pada halaman *form* jadwal

fisioterapi. Kemudian hasil *input* data akan dikirimkan pada kelas `C_Jadwal` menggunakan fungsi `tambah_jadwal()`. Selanjutnya data tersebut akan ditambah ke dalam *database* menggunakan fungsi `get_jadwal()` pada kelas `M_Jadwal`. Jika data pasien tersebut berhasil ditambahkan, maka *Admin* diarahkan menuju halaman jadwal fisioterapi.

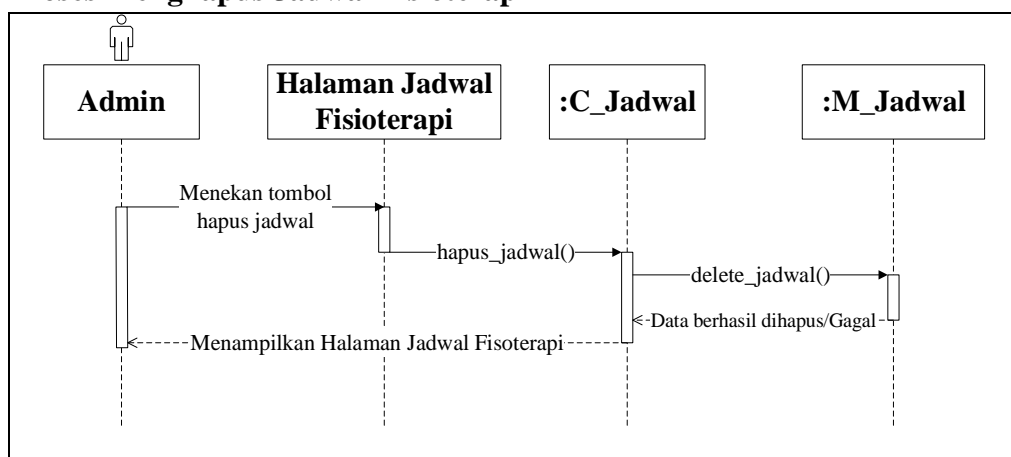
## 6. Proses Mengubah Jadwal Fisioterapi



Gambar 3.19 Proses mengubah jadwal fisioterapi.

Pada Gambar 3.19 merupakan proses mengubah jadwal fisioterapi, dimulai ketika *Admin* menekan tombol ubah jadwal fisioterapi pada halaman *form* ubah jadwal. Kemudian hasil *input* data terbaru akan dikirimkan ke kelas `C_Jadwal` menggunakan fungsi `ubah_jadwal()`. Selanjutnya data tersebut akan dikonfirmasi ke dalam *database* menggunakan fungsi `update_jadwal()` pada kelas `M_Jadwal`. Jika berhasil dilakukan perubahan, maka *Admin* diarahkan menuju halaman jadwal fisioterapi.

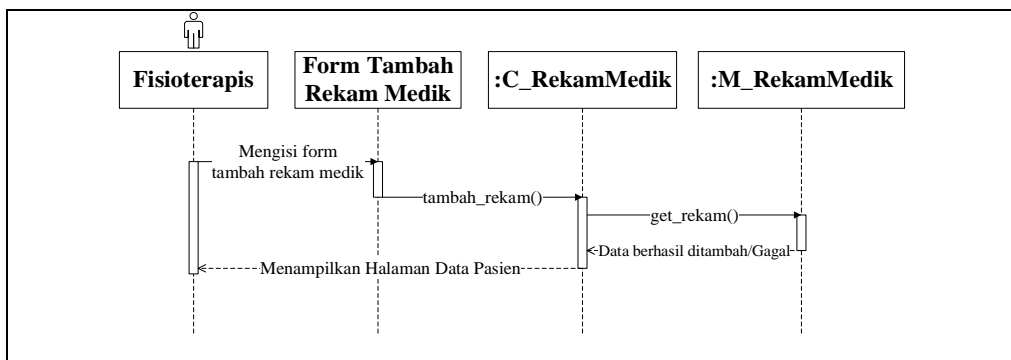
## 7. Proses Menghapus Jadwal Fisioterapi



Gambar 3.20 Proses menghapus jadwal fisioterapi.

Pada Gambar 3.20 merupakan proses menghapus jadwal fisioterapi, dimulai ketika *Admin* menekan tombol hapus jadwal pada halaman jadwal fisioterapi. Kemudian dilakukan konfirmasi penghapusan jadwal fisioterapi pada kelas *C\_Jadwal* menggunakan fungsi *hapus\_jadwal()*. Selanjutnya data tersebut akan dihapus dari dalam *database* menggunakan fungsi *delete\_jadwal()* pada kelas *M\_Jadwal*. Jika data pasien berhasil dihapus, maka *Admin* diarahkan menuju halaman data pasien.

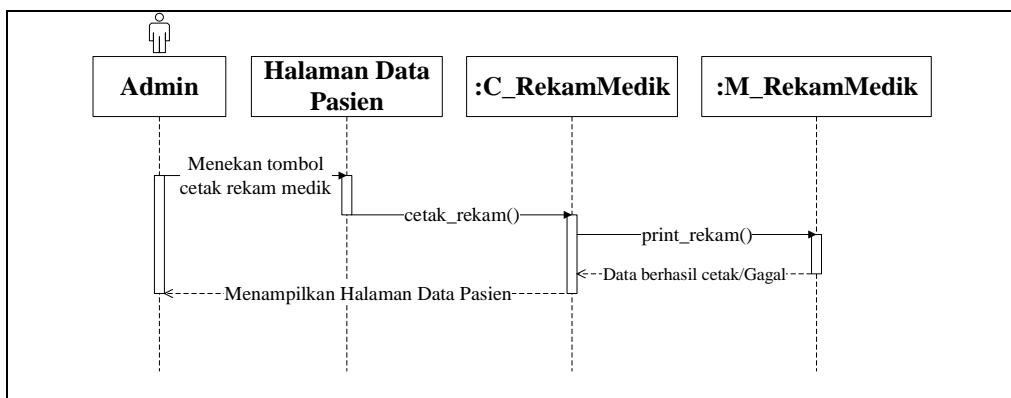
### 8. Proses Menambah Rekam Medik



Gambar 3.21 Proses menambah rekam medik.

Pada Gambar 3.21 merupakan proses menambah rekam medik, dimulai ketika Fisioterapis mengisi rekam medik baru pada halaman *form* tambah rekam medik. Kemudian hasil *input* data akan dikirimkan pada kelas *C\_RekamMedik* menggunakan fungsi *tambah\_rekam()*. Selanjutnya data tersebut akan ditambah ke dalam *database* menggunakan fungsi *get\_rekam()* pada kelas *M\_RekamMedik*. Jika rekam medik tersebut berhasil ditambahkan, maka Fisioterapis diarahkan menuju halaman data pasien.

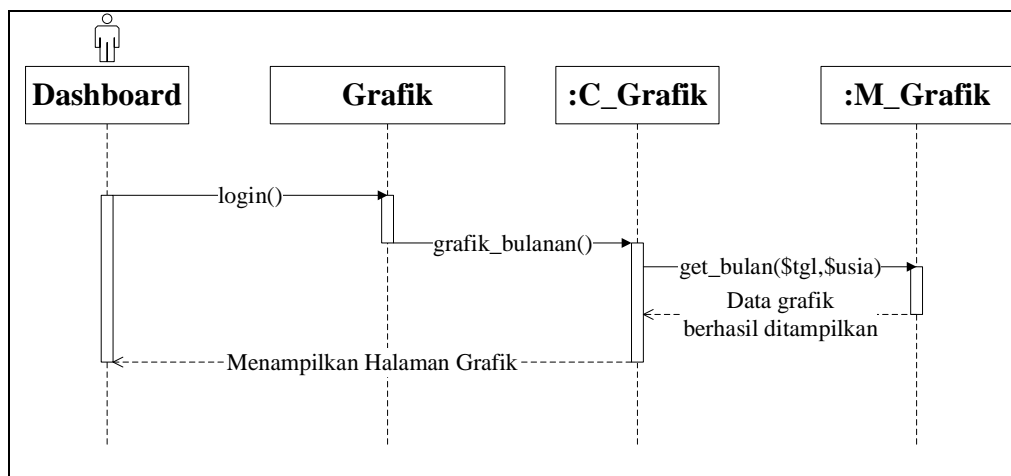
### 9. Proses Mencetak Rekam Medik



Gambar 3.22 Proses mencetak rekam medik.

Pada Gambar 3.22 merupakan proses mencetak rekam medik, dimulai ketika *Admin* menekan tombol cetak rekam medik pada halaman data pasien. Kemudian akan dikonfirmasi pada kelas `C_RekamMedik` menggunakan fungsi `cetak_rekam()`. Selanjut data tersebut akan dicetak sesuai dengan *database* menggunakan fungsi `print_rekam()` pada kelas `M_RekamMedik`. Jika rekam medik tersebut berhasil di *print*, maka *Admin* diarahkan menuju halaman data pasien.

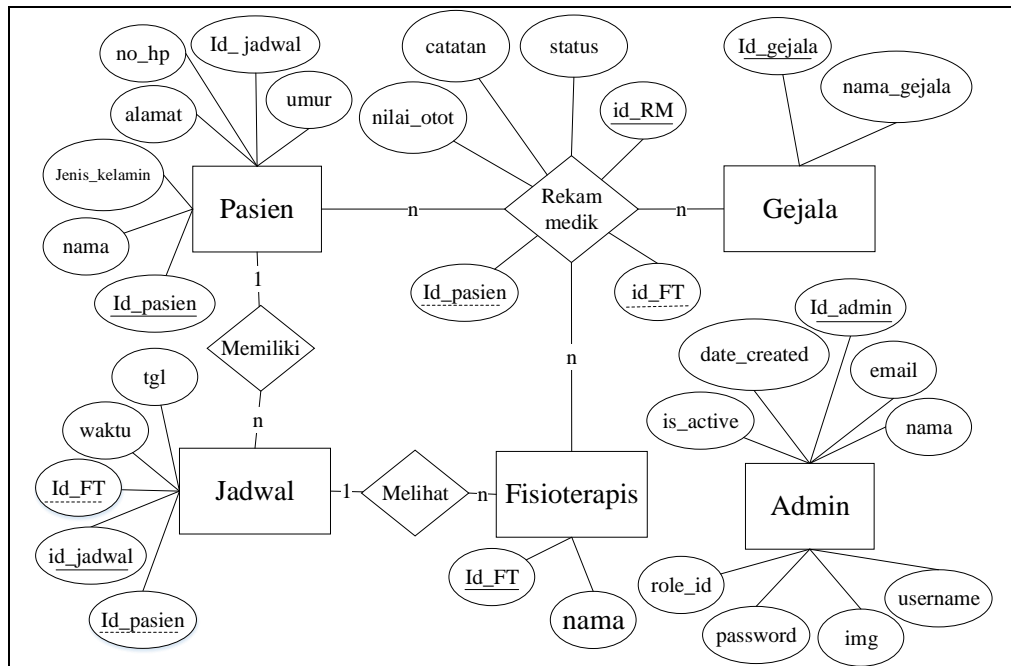
### 10. Proses Melihat Grafik Pasien



Gambar 3.23 Proses melihat grafik pasien.

Pada Gambar 3.23 merupakan proses melihat grafik pasien, dimulai ketika *Admin* melakukan *login* pada halaman *dashboard*. Kemudian akan dipanggil fungsi `grafik_bulanan()` pada kelas `C_Grafik`. Selanjut akan ditampilkan data pasien setiap bulan sesuai dengan *database* menggunakan fungsi `get_bulan()` pada kelas `M_Grafik`. Jika data pasien berhasil dipanggil, maka data pasien akan ditampilkan pada halaman *Dashboard* berupa grafik.

#### 4. Desain Database



Gambar 3.24 Desain database.

Berdasarkan Gambar 3.24 yang merupakan desain database didapatkan tabel-tabel database sebagai berikut:

##### 1. Tabel admin

Tabel *admin* merupakan tabel untuk menampung data pengguna yang melakukan *login* pada sistem.

Tabel 3.7 Tabel *admin*

No	Field Name	Type	Width
1	id_admin	Varchar	5
2	nama	Varchar	30
3	email	Varchar	30
4	img	Varchar	100
5	password	Varchar	30
6	role_id	Int	5
7	is_active	Int	11
8	date_created	Date	

##### 2. Tabel pasien

Tabel pasien merupakan tabel yang digunakan untuk menampung data pasien.

Tabel 3.8 Tabel pasien

No	Field Name	Type	Width
1	id_pasien	Varchar	5

2	nama	Varchar	30
3	alamat	Varchar	30
4	jenis_kelamin	Boolean	2
5	no_hp	Varchar	12
6	umur	Int	11
7	id_jadwal	Vachar	11

### 3. Tabel fisioterapis

Tabel fiiosterapis merupakan tabel yang digunakan untuk menampung data fisioterapis.

Tabel 3.9 Tabel fisioterapis

No	Field Name	Type	Width
1	id_FT	Varchar	5
2	nama	Varchar	30

### 4. Tabel jadwal

Tabel jadwal merupakan tabel yang digunakan untuk menampung data jadwal fisioterapi.

Tabel 3.10 Tabel jadwal

No	Field Name	Type	Width
1	id_jadwal	Varchar	5
2	tgl	Date	
3	waktu	Time	
4	Id_FT	Varchar	5
5	Id_pasien	Varchar	5

### 5. Tabel gejala

Tabel gejala merupakan tabel yang digunakan untuk menampung gejala pasien penyakit Bell's palsy.

No	Field Name	Type	Width
1	id_gejala	Varchar	5
2	nama_gejala	Varchar	30



#### 6. Tabel rekam medik

Tabel rekam medik merupakan tabel yang digunakan untuk menampung data rekam medik pasien Bell's palsy.

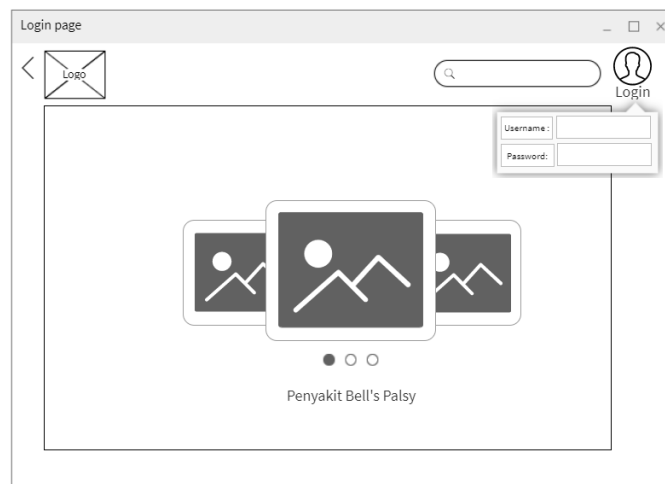
Tabel 3.11 Tabel rekam medik

No	Field Name	Type	Width
1	id_RM	Varchar	5
2	status	Varchar	7
3	nilai_otot	Varchar	5
4	catatan	Varchar	30
5	id_FT	Varchar	5
6	id_pasien	Varchar	5
7	id_gejala	Varchar	5

#### 3.4.2.3 Implementasi

Berikut merupakan rancangan pembuatan *design prototype* berupa *interface* dari sistem yang akan dibuat.

##### a. Halaman *login*

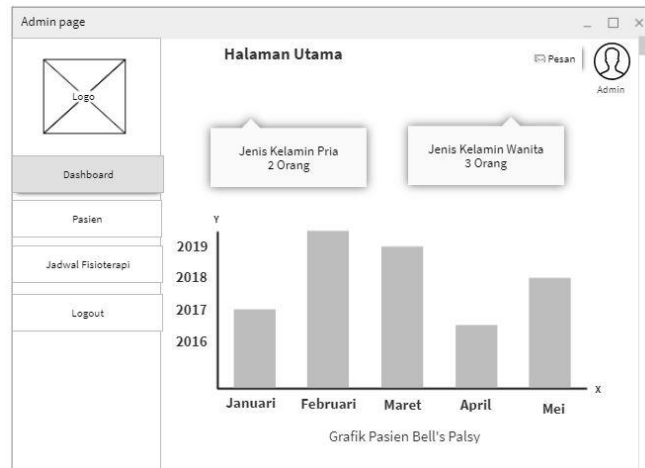


Gambar 3.25 Halaman *login Admin, Fisioterapis, dan Pasien.*

Pada Gambar 3.25 merupakan halaman awal untuk melakukan *login user admin, fisioterapis dan pasien* untuk masuk ke dalam sistem.

b. Halaman *admin*

1) Halaman utama *admin*



Gambar 3.26 Halaman utama *Admin*.

Pada Gambar 3.26 merupakan halaman utama pada *user Admin* yang dapat melihat grafik pasien selama satu tahun.

2) Halaman data pasien

No	Nama	Aksi	Rekam Medik
1		[Delete] [Edit] [Print]	[Print]
2		[Delete] [Edit] [Print]	[Print]
3		[Delete] [Edit] [Print]	[Print]

Gambar 3.27 Halaman data pasien.

Pada Gambar 3.27 merupakan halaman data pasien bagi *user admin*, dimana pada halaman data pasien *admin* dapat melakukan tambah data pasien baru, menghapus data pasien, merubah data pasien dan mencetak rekam medik pasien.

### 3) Halaman *form* tambah pasien

Admin page

Logo

Dashboard

Pasien

Jadwal Fisioterapi

Logout

Data pasien Bell's palsy

Pesan Admin

Form Tambah Pasien

Nama Pasien :

Alamat :

Umur :

Jenis Kelamin :  Laki-laki  Perempuan

No Hp :

Simpan Kembali

Gambar 3.28 *Form* tambah pasien.

Pada Gambar 3.28 merupakan tampilan *form* tambah pasien untuk menambah data pasien baru.

### 4) Halaman jadwal fisioterapi

Admin page

Logo

Dashboard

Pasien

Jadwal Fisioterapi

Logout

Jadwal Fisioterapi Bell's palsy

Pesan Admin

Jadwal Fisioterapi

Kalender Bulan Februari

Libur

Full

Less

Tambah jadwal

No	tgl	Waktu	Nama	Aksi
1				
2				
3				

Gambar 3.29 Halaman jadwal fisioterapi.

Pada Gambar 3.29 merupakan halaman jadwal fisioterapi yang dapat dilakukan oleh *user admin*, dimana pada halaman jadwal fisioterapi *admin* dapat melakukan tambah jadwal fisioterapi, menghapus jadwal dan merubah jadwal.

5) Halaman *form* tambah jadwal fisioterapi

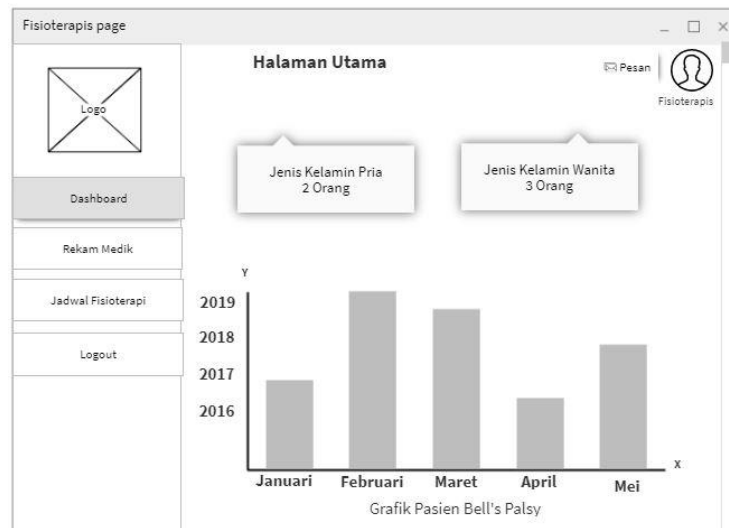
The screenshot shows a web application window titled 'Admin page'. On the left is a sidebar menu with options: 'Dashboard', 'Pasien', 'Jadwal Fisioterapi' (highlighted), and 'Logout'. The main content area is titled 'Jadwal Fisioterapi Bell's palsy' and contains a 'Form Jadwal Fisioterapi'. The form has four input fields: 'Id pasien', 'Nama pasien', 'Tanggal', and 'Waktu'. Below the form are two buttons: 'Simpan' and 'Kembali'. The top right corner shows a 'Pesanan' icon and a user profile for 'Admin'.

Gambar 3.30 *Form* tambah jadwal fisioterapi.

Pada Gambar 3.30 merupakan tampilan *form* tambah jadwal fisioterapi untuk menambah jadwal baru baru.

c. Halaman fisioterapis

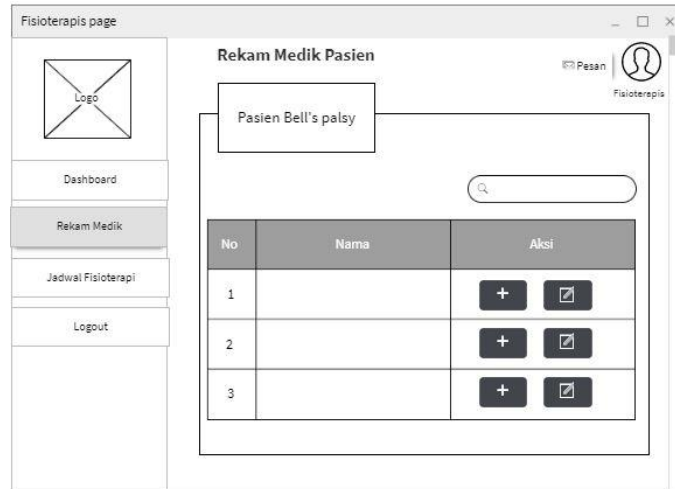
1) Halaman utama fisioterapis



Gambar 3.31 Halaman utama fisioterapis.

Pada Gambar 3.31 merupakan tampilan awal atau utama untuk user fisioterapis. Halaman utama ini menampilkan data pasien yang ditangani selama satu tahun.

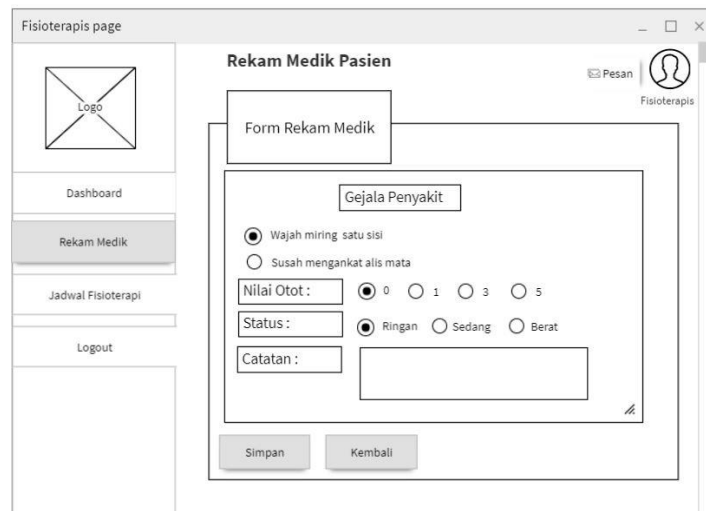
## 2) Halaman rekam medik



Gambar 3.32 Halaman rekam medik.

Pada Gambar 3.32 merupakan halaman rekam medik yang dapat dilakukan oleh *user* fisioterapis, dimana pada halaman rekam medik fisioterapis dapat melakukan tambah rekam medik sesuai dengan nama pasien, dan merubah rekam medik.

## 3) Halaman *form* tambah rekam medik

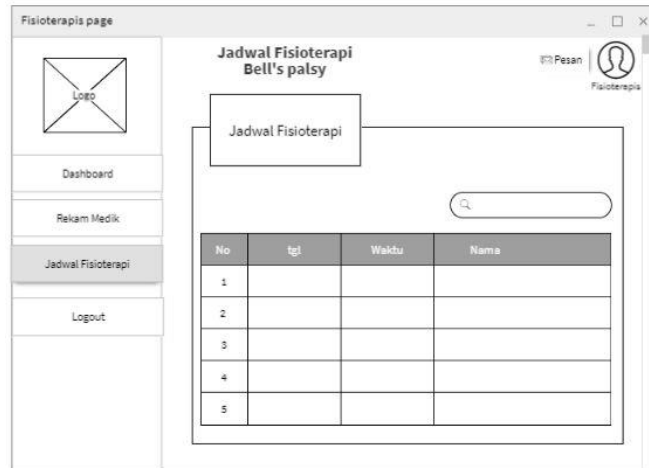


Gambar 3.33 Halaman *form* tambah rekam medik.

Pada Gambar 3.33 merupakan halaman *form* tambah rekam medik sesuai dengan data pasien, dimana fisioterapis melakukan tahapan mulai dari

gejala penyakit Bell's palsy, kemudia nilai otot pasien dan jika ada tambahan catatan penting untuk pasien.

4) Halaman jadwal fisioterapis

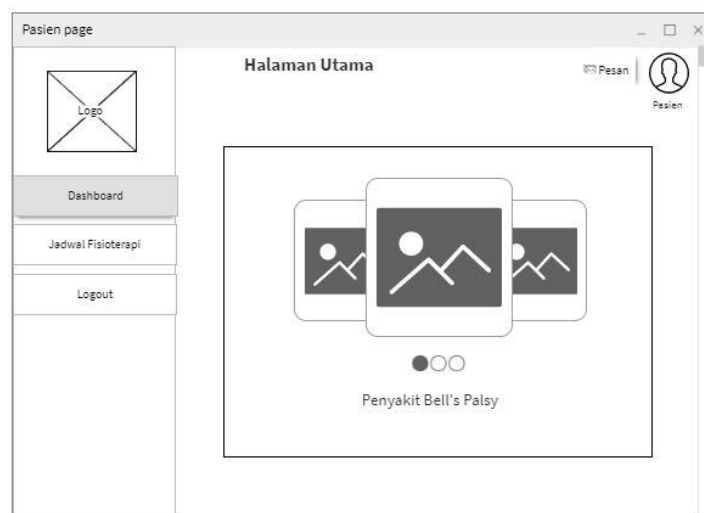


Gambar 3.34 Halaman jadwal fisioterapis.

Pada Gambar 3.34 merupakan tampilan halaman jadwal fisioterapis pada *user* fisioterapis, dimana fisioterapis hanya dapat melihat jadwal fisioterapis sesuai dengan kolom jadwal fisioterapi.

d. Halaman pasien

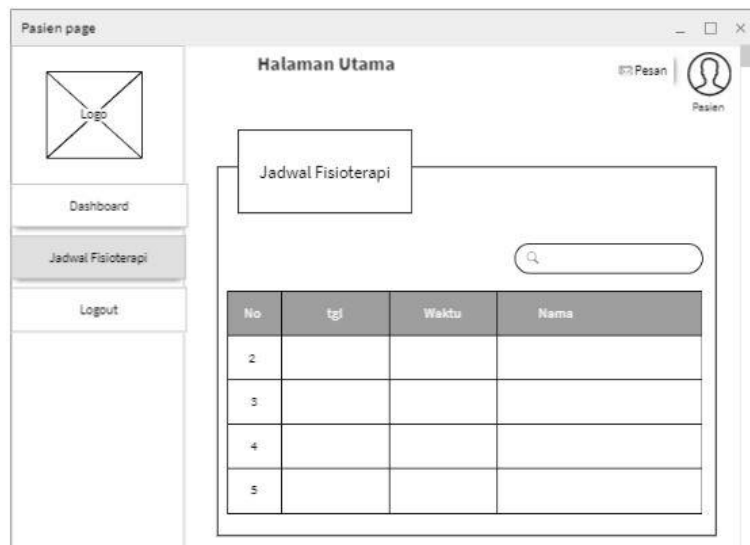
1) Halaman utama pasien



Gambar 3.35 Halam utama pasien.

Pada Gambar 3.35 merupakan tampilan utama untuk *user* pasien, dengan menampilkan informasi penyakit Bell's palsy.

## 2) Halaman jadwal pasien



Gambar 3.36 Halaman jadwal pasien.

Pada Gambar 3.36 merupakan tampilan halaman jadwal pasien, yaitu berfungsi untuk melihat jadwal fisioterapi Bell's palsy.

### 3.4.2.4 Test Elaboration

*Test elaboration* merupakan tahapan pengujian fitur yang telah dirancang pada *use case diagram* dengan implementasi desain *interface* sistem, pengujian dilakukan berdasarkan keinginan apakah sudah sesuai atau tidak. Pengujian dilakukan oleh pakar Fisioterapis dan *admin*. Proses validasi fitur dimulai dengan peneliti menampilkan rancangan *interface* sistem seperti pada Gambar 3.25 sampai Gambar 3.36, setelah itu penguji memberikan pernyataan setuju atau tidak setuju pada tabel isian yang telah disediakan. Hasil validasi dapat dilihat pada Tabel 3.12 berikut.

Tabel 3.12 Validasi fitur sistem.

No	Pengguna	Fitur	Validasi	
			Setuju	Tidak Setuju
1	Admin	Menambah data pasien	✓	
		Mengubah data pasien	✓	
		Menghapus data pasien	✓	
		Menambah jadwal	✓	

		Mengubah jadwal	✓	
		Menghapus jadwal	✓	
		Mencetak rekam medik	✓	
		Melihat grafik pasien	✓	
		Melihat jadwal	✓	
2	Fisioterapis	Melihat grafik pasien	✓	
		Melihat rekam medik	✓	
		Menambah rekam medik	✓	
		Melihat jadwal	✓	

### 3.4.3 Tahap *construction*

Tahap *construction* merupakan tahap implementasi sistem yang mulai dibangun. Adapun tahapannya yaitu sebagai berikut.

- a. Membangun antar muka atau *interface* sistem menggunakan *framework codeigniter*.
- b. Pembuatan *database* sistem menggunakan MySQL.
- c. Pengkodean sistem dengan menggunakan konsep *Model View Controller* (MVC).
- d. Pengujian sistem oleh masing-masing aktor untuk mengetahui apakah tampilan dan fungsional sistem telah sesuai dengan kebutuhan.

### 3.4.4 Tahap *transition*

Tahap *transition* adalah tahap terakhir dalam metode RUP, dimana pada tahap ini dilakukan beberapa aktivitas yaitu sebagai berikut.

- a. Pengujian *black box*

Pengujian *black box* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti melihat suatu kotak hitam, hanya penampilan luarnya saja yang bisa terlihat, tanpa mengetahui ada apa dibalik bungkus hitamnya. *Black box testing* melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan *output*. Pengujian ini juga dapat mengukur



kemampuan sistem dalam memenuhi kebutuhan pengguna serta mengetahui kesalahan-kesalahan pada sistem berdasarkan keluaran yang dihasilkan. Pengujian *black box* pada penelitian ini melibatkan 5 orang responden (mahasiswa) untuk menguji apakah fitur yang ada dalam sistem telah sesuai dengan yang diharapkan.

b. Pengujian *Mean Opinion Score* (MOS)

Pengujian MOS dilakukan dengan tujuan memperoleh tanggapan pengguna untuk mengetahui apakah sistem yang dibuat mudah untuk digunakan oleh pengguna, fitur-fitur yang ada dalam sistem sudah berjalan dengan baik, maupun kemampuan sistem yang dapat memberikan informasi mengenai gangguan mental pada anak dengan baik. Pengujian ini dilakukan dengan menggunakan kuesioner yang melibatkan 30 orang responden yang terdiri dari 15 mahasiswa, 10 masyarakat umum, dan 5 orang tenaga medis sesuai dengan ilmu penerapan masing-masing, dimana responden akan memberi penilaian terhadap sistem kemudian hasil penilaian akan dihitung dengan parameter MOS untuk mendapat kesimpulan hasil pengujian

c. Sistem yang telah selesai dilakukan pengujian dan telah sesuai dengan kebutuhan akan diserahkan kepada instansi untuk dapat digunakan sebagaimana mestinya.

### 3.4.5 Jadwal Kegiatan

Waktu yang akan digunakan selama proses pengembangan sistem informasi konsultasi dan penjadwalan yaitu selama enam bulan. Jadwal kegiatan dalam pengembangan sistem pakar dapat dilihat pada Tabel 3.13.

Tabel 3.13

No.	Kegiatan	Waktu (bulan)							Keterangan
		I	II	III	IV	V	VI	VII	
1.	<i>Identification</i>								Analisis sistem
2.	<i>Requirements</i>								Perancangan sistem
3.	<i>Coding</i>								Pengkodean sistem
4.	<i>Testing</i>								Pengujian sistem
5.	<i>Dokumentation</i>								Dokumentasi sistem

6.	<i>Deployment</i>									Instalasi sistem untuk <i>user</i>
----	-------------------	--	--	--	--	--	--	--	--	------------------------------------

## DAFTAR PUSTAKA

- [1] H. H. Bin Abdullah, “Karakteristik Pasien Bell’s Palsy Di Rumah Sakit Umum Pusat Dr. Wahidin Sudirohusodo Pada Tahun 2016,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2017.
- [2] P. S. Hasibuan and M. I. Batubara, “Penerapan Metode Dempster Shafer Dalam Mendiagnosa Penyakit Faringitis,” *Muhammad Syaril, Nelly Astuti Hasibuan, Pristiwanto*, vol. 3, no. 1, pp. 101–105, 2016.
- [3] O. M. Adam, “Bell ’ s Palsy,” *Library (Lond).*, vol. 2071, no. 1, pp. 1323–1331, 2019.
- [4] W. R. Amanda, “Penatalaksanaan Fisioterapi Untuk Meningkatkan Kekuatan Otot Dan Mengembalikan Kesimetrisan Wajah Pada Kasus Bell’s Palsy Sinistra Di Rsud Dr. Soeselo Slawi,” *Fisioter. Fak. Ilmu Kesehat.*, 2019.
- [5] F. Mubarak, H. Harliana, and I. Hadijah, “Perbandingan Antara Metode RUP dan Prototype Dalam Aplikasi Penerimaan Siswa Baru Berbasis Web,” *Creat. Inf. Technol. J.*, vol. 2, no. 2, p. 114, 2015.
- [6] M. Destiningrum and Q. J. Adrian, “Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre),” *J. Teknoinfo*, vol. 11, no. 2, p. 30, 2017.
- [7] H. Tohari, *Astah - Analisis serta perancangan sistem Informasi melalui pendekatan UML*. Yogyakarta: ANDI, 2014.
- [8] M. Faizal and S. L. Putri, “Sistem Informasi Pengolahan Data Pegawai Berbasis Web (Studi Kasus Di Pt Perkebunan Nusantara Viii Tambaksari),” *J. Teknol. Inf. dan Komun. STMIK Subang*, pp. 1–23, 2017.
- [9] R. Taufiq, *Sistem Informasi Manajemen*. Yogyakarta: Graha Ilmu, 2013.
- [10] S. K. Boell and D. Cecez-Kecmanovic, “What is an information system?,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2015–March, no. March, pp. 4959–4968, 2015.
- [11] J. Hartono, *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset, 2005.

- [12] T. Sutabri, *Analisis Sistem Informasi*. Yogyakarta, 2012.
- [13] H. Hartono, “Pengertian Website Dan Fungsinya,” *Ilmu Teknol. Inf.*, pp. 2–7, 2014.
- [14] P. Adriansyah, “Mengenal Framework Codeigniter,” 2017.
- [15] H. Alatas, *Responsive Web Design dengan PHP dan Bootstrap*. Yogyakarta: Lokomedia, 2013.
- [16] H. Saputro, “MODUL PEMBELAJARAN PRAKTEK BASIS DATA ( MySQL ),” *Modul Pembelajaran Prakt. Basis Data*, pp. 1–34, 2016.
- [17] B. Raharjo, *Belajar Otodidak Membuat Database Menggunakan MySQL*. Bandung: Informatika, 2011.
- [18] M. K. Ir. Yuni Sugiarti, “Analisis & Perancangan UML (Unified Modeling Language) Generated VB.6,” *Graha Ilmu*, pp. 30–45, 2013.
- [19] S. J. Falchek, “Bell palsy,” *5-Minute Pediatr. Consult. 8th Ed.*, no. December, pp. 100–101, 2018.
- [20] A. Soliman, A. Mettleq, I. M. Dheir, A. A. Elsharif, and S. S. Abu-naser, “Expert System for the Diagnosis of Seventh Nerve Inflammation ( Bell ’ s palsy ) Disease,” vol. 3, no. 4, pp. 27–35, 2019.
- [21] S. Young, “Penerapan Metode SDLC RUP (Rational Unified Process) Dalam Pembuatan Sistem Informasi E-Kos Berbasis Web Pada Kota Malang,” *Univ. Muhammadiyah Malang*, 2018.

## LAMPIRAN

### LEMBAR PENGUJIAN VALIDASI KEBUTUHAN FUNSIONAL SISTEM DENGAN USE CASE DALAM PENGEMBANGAN SISTEM INFORMASI DAN KONSULTASI PENYAKIT BELL'S PALSY PADA KLINIK GRIYA FISIOTERAPIS

#### D. Identitas Responden

Nama : P. Arif Trianto C, S.P.T, FT, S.Th.I., NMTCF-SM  
Pekerjaan : Fisioterapist  
Instansi : Klinik Griya Fisioterapi

#### E. Petunjuk Pengisian

Berilah tanda centang (✓) pada kolom Admin jika kebutuhan fungsional sistem sesuai dengan use case admin, Fisioterapis jika kebutuhan fungsional sistem sesuai dengan use case fisioterapis dan begitu juga dengan Pasien.

#### F. Lembar pertanyaan

No	Kebutuhan fungsional sistem	Responden		
		Admin	Fisioterapis	Pasien
1	Mengelola pasien (menambah, mengubah, dan menghapus)	✓		
2	Mengelola jadwal (menambah, mengubah, dan menghapus)	✓		
3	Mencetak rekam medik	✓		
4	Melihat grafik pasien	✓	✓	
5	Melihat rekam medik		✓	
6	Menambah rekan medik		✓	
7	Melihat jadwal	✓	✓	✓

Berdasarkan hasil dari proses validasi yang dilakukan, didapatkan kesimpulan bahwa kebutuhan fungsional sistem yang dirancang sesuai / tidak sesuai dengan kebutuhan pengguna.

Mataram, 11 Mei 2020  
Responden

P. Arif Trianto C, S.P.T, FT, S.Th.I., NMTCF-SM

**LEMBAR PENGUJIAN VALIDASI FITUR SISTEM DENGAN IMPLEMENTASI  
DESAIN DALAM PENGEMBANGAN SISTEM INFORMASI DAN KONSULTASI  
PENYAKIT BELL'S PALSY PADA KLINIK GRIYA FISIOTERAPIS**

**A. Identitas Responden**

Nama : R. Arif Trianto G.S.ST.FT., S.Th.I., NMTCP-8M  
 Pekerjaan : Fisioterapist  
 Instansi : Klinik Griya Fisioterapi

**B. Petunjuk Pengisian**

Berilah tanda centang (✓) pada kolom **Setuju** jika fitur sistem dengan kebutuhan dan **Tidak Setuju** jika fitur sistem tidak sesuai dengan kebutuhan.

**C. Lembar pertanyaan**

No	Penggunaan	Fitur	Responden	
			Iya	Tidak
1	Admin	Menambah data pasien	✓	
2		Mengubah data pasien	✓	
3		Menghapus data pasien	✓	
4		Mencetak rekam medik	✓	
5		Melihat grafik pasien	✓	
6		Menambah jadwal fisioterapi	✓	
7		Mengubah jadwal fisioterapi	✓	
8		Menghapus jadwal fisioterapi	✓	
9		Melihat jadwal fisioterapi	✓	
10	Fisioterapis	Melihat grafik pasien	✓	
11		Melihat rekam medik	✓	
12		Melihat jadwal fisioterapi	✓	
13		Menambah rekam medik	✓	
14	Pasien	Melihat jadwal	✓	

Berdasarkan hasil dari proses validasi yang dilakukan, didapatkan kesimpulan bahwa fitur sistem yang dirancang sesuai / tidak sesuai dengan kebutuhan pengguna.

Mataram, 11 Mei ..... 2020  
 Responden

R. Arif Trianto G.S.ST.FT., S.Th.I., NMTCP-8M

**FORM PERTANYAAN KONSULTASI AWAL PASIEN BELL'S PALSY  
PADA KLINIK GRIYA FISIOTERAPIS**

**A. Identitas Responden**

Nama : R. Anif Trianto G.P.P.T.F.T., S.Th.I., NMTCF-SM  
 Pekerjaan : Fisioterapist  
 Instansi : Klinik Griya Fisioterapi

**B. Lembar pertanyaan**

No	Pertanyaan konsultasi penyakit Bell's palsy	Responden	
		Iya	Tidak
1	Sudah berapa lama penyakit yang anda alami?	✓	
2	Apakah mata susah tertutup atau merasa berat?	✓	
3	Apakah merasa mulut tidak simetris ke sebelah kiri atau kekanan?	✓	
4	Apakah Anda sulit untuk mengangkat alis mata?	✓	
5	Apakah Anda mengalami kesulitan saat berkumur (tumpah)?	✓	
6	Apakah makanan saat Anda makan berkumpul pada satu sisi kiri atau kanan?	✓	
7	Apakah ketika Anda meludah tidak simetris atau miring?	✓	
8	Apakah lidah miring ke satu sisi ketika dijulurkan?	✓	
9	Apakah Anda kurang merasakan sesuatu (Hiposensitif) di seputran lidah atau disekitar mulut?	✓	
10	Apakah Anda merasa nyeri kepala sebelah kanan atau kiri (dari depan kepala sampai dengan tengkuk disisi yang dirasakan sakit)?	✓	
11	Apakah Anda merasa nyeri tengkuk dibagian belakang telinga (nyeri berdenyut)?	✓	
12	Apakah Anda sering terpapar angin pada telinga?	✓	
13	Apakah Anda sering mengorek telinga hingga bagian dalam telinga?	✓	
14	Apakah Anda pernah mengalami benturan di daerah bagian telinga beberapa hari sebelum merasakan penyakit Bell's palsy?	✓	
15	Apakah Anda merasakan rasa gatal terbakar disertai luka melepuh di area telinga hingga tengkuk?	✓	

Berdasarkan hasil dari proses validasi yang dilakukan, didapatkan kesimpulan bahwa pertanyaan konsultasi awal yang dirancang sesuai / tidak sesuai dengan kebutuhan pengguna.

Mataram, 11 Mei 2020  
 Responder

R. Anif Trianto G.P.P.T.F.T., S.Th.I., NMTCF-SM

# [ta] Editor Decision

Kotak Masuk x



**NDIYASARI AGITHA** <if@unram.ac.id>

kepada saya ▾

Sen, 11 Mei 19.28 (6 hari yang lalu)



Rejalana Bayu Kurniawan Kurniawan:

We have reached a decision regarding your submission to Publikasi Tugas Akhir S-1 PSTI FT-UNRAM, "Rancang Bangun Sistem Informasi Dan Konsultasi Penjadwalan Fisioterapi Penyakit Bell's Palsy (Studi Kasus Klinik Griya Fisioterapis)".

Our decision is to: Accept Submission

Dipersilakan menggunakan laporan versi terakhir untuk mendaftar Seminar TA1.

Terima kasih,  
NADIYASARI AGITHA  
[nadiya@unram.ac.id](mailto:nadiya@unram.ac.id)

---

[Publikasi Tugas Akhir S-1 PSTI FT-UNRAM](#)

↩ Balas

➡ Teruskan