

BAB II

LANDASAN TEORI

2.1 Pelayanan Kesehatan

2.1.1 Puskesmas

Puskesmas atau Pusat Kesehatan Masyarakat adalah fasilitas pelayanan kesehatan yang menyelenggarakan upaya kesehatan masyarakat dan upaya kesehatan perseorangan tingkat pertama dengan lebih mengutamakan upaya promotif dan preventif untuk mencapai derajat kesehatan masyarakat yang setinggi-tingginya di wilayah kerjanya (Permenkes Nomor 43 Tahun 2019).

2.1.2 Rekam Medis

2.1.1 Pengertian Rekam Medis

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 269 (Permenkes, 2008), pengertian rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Catatan-catatan tersebut adalah tulisan yang dibuat oleh dokter atau dokter gigi mengenai seluruh tindakan yang telah diberikan kepada pasien dalam upaya pemberian pelayanan kesehatan.

Gemala Hatta (2008) rekam medis adalah kumpulan fakta tentang kehidupan seseorang dan riwayat penyakitnya termasuk keadaan sakit, pengobatan saat ini dan saat lampau yang ditulis oleh para praktisi kesehatan dalam upaya (mereka) memberikan upaya pelayanan kepada pasien.

Menurut (Permenkes Nomor 24, 2022) pengertian rekam medis elektronik adalah rekam medis yang dibuat dengan menggunakan sistem elektronik yang diperuntukkan bagi penyelenggaraan rekam medis.

2.1.2 Kegunaan Rekam Medis

Kegunaan rekam medis menurut (Yanmed, 2006), dapat dilihat dari beberapa aspek diantaranya sebagai berikut:

A. Aspek Adminitrasi (*Administration*)

Seluruh data dan informasi yang dihasilkan dalam rekam medis mempunyai nilai administrasi karena menyangkut tindakan berdasarkan wewenang dan tanggung jawab sebagai tenaga medis dan paramedis dalam mencapai tujuan pelayanan kesehatan.

B. Aspek Hukum (*Legal*)

Seluruh data dan informasi dalam rekam medis mempunyai nilai hukum, karena menyangkut masalah adanya jaminan kepastian hukum atas keadilan selain itu, dalam rangka menegakkan hukum serta penyediaan bahan tanda bukti untuk menegakkan keadilan.

C. Aspek Keuangan (*Financial*)

Seluruh data dan informasi dalam rekam medis mempunyai nilai finansial, karena data dan informasi dalam rekam medis dapat dijadikan sebagai bahan untuk menetapkan biaya pembayaran pelayanan kesehatan.

D. Aspek Penelitian (*Research*)

Seluruh data dan informasi dalam rekam medis mempunyai nilai riset karena data dan informasi dalam rekam medis dapat digunakan sebagai aspek penelitian

dan pengembangan ilmu pengetahuan tepatnya pada bidang kesehatan.

E. Aspek Pendidikan (*Education*)

Seluruh data data dan informasi dalam rekam medis mempunyai nilai pendidikan karena data dan informasi di dalam rekam medis memuat perkembangan kronologis dan kegiatan pelayanan medis yang diberikan kepada pasien, dari informasi tersebut dapat digunakan sebagai referensi pembelajaran dibidang profesi pemakai.

F. Aspek Dokumentasi (*Documentation*)

Seluruh isi dalam rekam medis mempunyai nilai dokumentasi karena di dalamnya dapat digunakan sebagai sumber ingatan yang harus didokumentasikan dan digunakan sebagai bahan pertanggung jawaban dan laporan rumah sakit.

2.1.3 Standar Operasional Prosedur

Pengertian Standar Operasional Prosedur Berdasarkan Peraturan Pemerintah Kesehatan Indonesia Nomor 512 (Kemenkes, 2007), standar prosedur operasional adalah suatu perangkat instruksi atau langkah-langkah yang dibakukan untuk menyelesaikan suatu proses kerja rutin tertentu, dimana standar prosedur operasional memberikan langkah yang benar dan terbaik berdasarkan konsensus bersama untuk melaksanakan berbagai kegiatan dan fungsi pelayanan yang dibuat oleh sarana pelayanan kesehatan berdasarkan standar profesi.

2.2 *User Interface*

User interface atau UI adalah istilah yang digunakan untuk menggambarkan tampilan pada sebuah *website* atau komputer yang berinteraksi langsung dengan pengguna. UI bukan hanya sekedar tentang

warna dan bentuk, melainkan tentang menyajikan *tools* (alat) yang tepat kepada *user* (pengguna) untuk mencapai tujuan utama dalam UI. Dalam sistematisa fungsi UI tidak hanya sekedar tombol (*button*) dan menu yang harus dilakukan oleh *user*, namun lebih kepada bagaimana sebuah UI dapat berinteraksi dengan pengguna dalam sebuah aplikasi. Hal ini menjelaskan bahwa UI bukan hanya sekedar tampilan produk tetapi tentang bagaimana produk tersebut dapat bekerja. Dalam desain UI seorang designer harus mampu menjaga keseimbangan antara estetika dan interaktivitas tanpa memerlukan usaha ekstra (Rochmawati, 2019).

2.3 Sistem Android

Android adalah sebuah sistem operasi pada perangkat mobile berbasis *linux* yang didalamnya mencakup sistem operasi *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para *developer* untuk mengembangkan aplikasi mereka. Sistem operasi android ini digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan *tablet* (Safitri & Basuki, 2020).

2.4 Prototype

Menurut Raymond McLeod dalam jurnal (Mulyanto & Gunawan, 2021), *Prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial terkait cara sistem berfungsi dalam bentuk lengkap dan proses untuk menghasilkan sebuah *Prototype* adalah *prototyping*. *Prototype* adalah gambaran hampir menyerupai bentuk asli dari sebuah sistem yang dapat diubah sesuai dengan kebutuhan sebelum direalisasikan sehingga dapat menekan biaya pengeluaran.

2.5 Konsep UML (*Unified Modelling Language*)

2.5.1 Pengertian UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu standart bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta

menggambarkan arsitektur dalam pemograman berorientasi objek (Hasanah & Untari, 2020).

Menurut Sugiarti dalam buku yang ditulis oleh (Anardani, 2019) UML atau *Unified Modelling Language* adalah alat bantu yang sudah menjadi standar dalam dunia pengembangan sistem perangkat lunak berorientasi objek.

Dari pengertian di atas, UML dapat diartikan sebagai suatu bahasa untuk memvisualisasikan atau menggambarkan definisi-definisi terkait *requirement*, membuat analisis dan desain, serta menggambarkan asitektur dalam pemograman berorientasi objek.

2.5.2 *Use case Diagram*

Menurut (Hasanah & Untari, 2020) *use case diagram* adalah sebuah teknik untuk merekam persyaratan fungsional sebuah sistem, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sedangkan maksud dari *use case* adalah gambaran fungsionalitas dari suatu sistem dengan tujuan pengguna sistem paham dan mengerti terkait kegunaan sistem yang akan dibangun. Dengan menggunakan *use case diagram* akan sangat membantu dalam menyusun *requirement* analisis atau menyusun kebutuhan pada sebuah sistem.

Use case diagram akan digunakan untuk menyambungkan rancangan sistem klien, serta merancang *test case* untuk keseluruhan *feature* yang ada pada sistem. Terdapat dua hal utama pada *use case* adalah pendefinisian dengan apa yang dimaksud dengan Aktor dan *use case*.

- A. Aktor adalah seseorang, proses atau sistem lain yang berinteraksi dengan sistem serta memungkinkan hanya memberikan informasi inputan pada sistem, menerima informasi dari sistem atau keduanya saling menerima

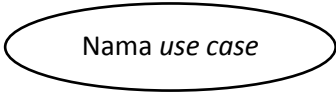
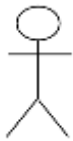
dan memberikan informasi pada sistem. Meskipun aktor digambarkan dengan simbol orang belum tentu aktor adalah orang.

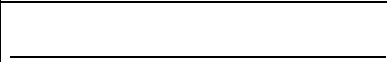

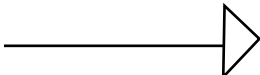

B. Use case

Use case adalah penggambaran atau visualisasi fungsionalitas suatu sistem, dilihat dari sudut pandang *user* (pengguna) atau aktor yang menitikberatkan pada fungsionalitas pada sistem bukan alur ataupun urutan kejadian sehingga *user* dapat memahami kegunaan sistem yang akan dirancang (Hasanah & Untari, 2020).

Simbol-simbol yang digunakan pada proses pembuatan *use case* diagram :

Tabel 2. 1 Simbol *Use case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Penggambaran atau visualisasi fungsionalitas suatu sistem untuk saling bertukar pesan antar aktor.</p>
<p>Aktor / <i>actor</i></p> 	<p>Seseorang, proses atau sistem lain yang berinteraksi dengan sistem serta memungkinkan hanya memberikan informasi inputan pada sistem, menerima informasi dari sistem atau keduanya saling menerima dan memberikan informasi pada sistem. Meskipun aktor digambarkan dengan simbol orang belum tentu aktor adalah orang.</p>
<p>Asosisasi / <i>assocoation</i></p>	<p>Sebuah relasi untuk menghubungkan antara aktor dan</p>


Simbol	Deskripsi
	<i>case</i> di dalam satu diagram.
Ekstensi / <i>extend</i> << <i>extend</i> >> 	Menjelaskan bahwa suatu <i>use case</i> dapat dijalankan karena terdapat syarat tertentu dari <i>use case</i> lain.
Generalisasi / <i>generalization</i> 	Relasi yang menyatakan hubungan turunan dalam satu komponen atau disebut juga dengan <i>inheritance</i> (pewarisan) sehingga sebuah elemen dapat menjadi spesialisasi dari elemen lain.
Menggunakan / <i>include</i> << <i>include</i> >> 	Menggambarkan tindakan yang harus dilakukan agar sebuah peristiwa dapat terjadi, kondisi ini adalah <i>use case</i> bagian dari <i>use case</i> lainnya.

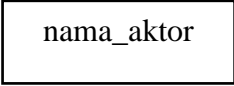

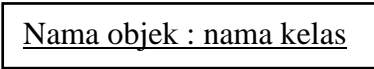



Sumber : (Shalahuddin & Rosa, 2019)



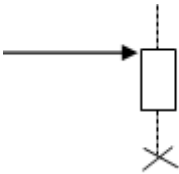
2.5.3 Sequence Diagram

Sequence diagram atau diagram sekuen adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* (pesan) yang dikirimkan dan diterima antar objek (Shalahuddin & Rosa, 2019). Simbol-simbol yang digunakan pada proses pembuatan *sequence diagram* :

Tabel 2. 2 Simbol *Sequence Diagram*

Simbol	Deskripsi
 Atau	Seseorang, proses atau sistem lain yang berinteraksi dengan sistem serta memungkinkan hanya memberikan informasi

Simbol	Deskripsi
	<p>inputan pada sistem, menerima informasi dari sistem atau keduanya saling menerima dan memberikan informasi pada sistem. Meskipun aktor digambarkan dengan simbol orang belum tentu aktor adalah orang.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menandakan kehidupan pada suatu objek</p>
<p>Objek</p> 	<p>Menjelaskan objek yang berinteraksi dengan mengirim atau menerima pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan bahwa objek pada keadaan aktif dan sedang berinteraksi atau objek mengirim atau menerima pesan.</p>
<p>Pesan tipe <i>create</i></p> <p><< <i>create</i> >></p> 	<p>Menyatakan suatu objek membuat objek yang lain arah panah, mengarah pada objek yang akan dibuat</p>
<p>Pesan tipe <i>call</i></p> <p>1:nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau diri sendiri</p>

Simbol	Deskripsi
Pesan tipe <i>send</i> 1: masukan 	Menyatakan suatu objek mengirimkan masukan/ data / informasi kepada objek lain
Pesan tipe <i>return</i> 1: keluaran 	Menyatakan suatu objek yang telah menjalankan suatu operasi/metode maka akan menghasilkan keluaran/ kembalian ke objek tertentu
Pesan tipe <i>destroy</i> << destroy >> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri

Sumber : (Shalahuddin & Rosa, 2019)

2.5.4 Class Diagram

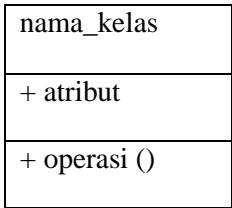
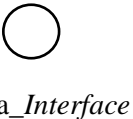


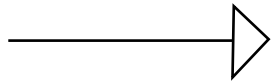

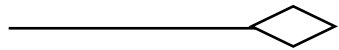
Class diagram atau diagram kelas adalah sebuah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Shalahuddin & Rosa, 2019). Diagram kelas berfungsi untuk membantu programmer dalam membuat kelas-kelas sesuai dengan rancangan diagram kelas agar antara dokumentasi pembuatan dan perangkat lunak sinkron.

Kelas sendiri memiliki atribut dan metode atau operasi, yaitu:

1. Atribut adalah variabel-variabel yang dimiliki oleh suatu kelas
2. Metode atau operasi merupakan fungsi yang dimiliki oleh suatu kelas

Simbol-simbol yang digunakan pada proses pembuatan *class* diagram:

Tabel 2. 3 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas (<i>class</i>)</p> 	<p>Kumpulan dari objek atau kelas yang berbagi atribut dan operasi yang sama pada struktur sistem</p>
<p>Antarmuka (<i>Interface</i>)</p> 	<p>Konsep <i>Interface</i> pada pemograman yang berorientasi pada objek</p>
<p>Asosiasi (<i>association</i>)</p> 	<p>Penghubung antar kelas secara umum, biasanya disertai dengan <i>multiplicity</i></p>
<p>Asosiasi berarah (<i>directed association</i>)</p> 	<p>Penghubung antar kelas dengan maksud kelas satu digunakan kelas lainnya, biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Generalisasi</p> 	<p>Hubungan antar kelas dengan maksud umum-khusus</p>
<p>Kebergantungan (<i>dependency</i>)</p> 	<p>Relasi antar kelas yang saling kebergantungan antar kelas</p>
<p>Agregasi (<i>aggregation</i>)</p> 	<p>Hubungan antar kelas dengan semua bagian</p>

Sumber : (Shalahuddin & Rosa, 2019)

2.6 Warna

Berdasarkan (Schlatter & Levinson, 2013) dalam memberikan panduan untuk menyusun sebuah desain aplikasi yang mudah dengan membaginya ke dalam beberapa komponen salah satunya adalah warna. Secara teori warna dibagi menjadi dua kategori diantaranya adalah warna additif dan warna substraktif. Warna additif adalah warna yang dihasilkan oleh cahaya, umumnya digunakan pada televisi atau layar komputer. Contoh warna primer dari warna additif adalah merah, biru, dan hijau. Sedangkan, warna substraktif adalah warna yang dihasilkan dari pigmen. Pigmen yang dimaksud seperti cat, pewarna, atau tinta. Contoh warna primer dari warna substraktif adalah *cyan* (biru), *magenta* (merah), dan kuning. Penggunaan warna pada *website* tidak hanya untuk membuat tampilan *website* menarik untuk dilihat namun juga untuk memberikan informasi, menghibur, serta membangkitkan perasaan bawah sadar pada pengguna (Powell, 2002).

2.7 Layout

Layout adalah susunan atau tata letak dari suatu elemen di dalam sebuah media tertentu seperti pada aplikasi atau komputer. Tujuan elemen-elemen dalam *layout* adalah untuk menyampaikan informasi dengan lengkap dan tepat kepada pengguna (Monica, 2010). Dalam desain, elemen-elemen pada *layout* digunakan untuk mengatur tata letak dalam suatu media, seperti situs *web* atau aplikasi. Keseluruhan elemen pada *layout* bekerja sama untuk menciptakan suatu tampilan yang terorganisir dan menarik. Beberapa elemen-elemen pada *layout* menurut (Schlatter & Levinson, 2013), diantaranya:

1. *Screen size* (ukuran layar). Sebelum membuat sebuah perlu diketahui terlebih dahulu ukuran yang akan digunakan supaya seluruh elemen dapat masuk dalam *frame* dan saling berhubungan.

2. *Position* (posisi). Didalam *layout* penempatan elemen perlu dipertimbangkan antar elemen lainnya untuk menciptakan tampilan yang sempurna.
3. *White space* (ruang kosong), penggunaan *white space* pada *layout* dapat membantu untuk memberikan ruang pada elemen yang berbeda.
4. *Proximity* (kedekatan), penggunaan *proximity* pada *layout* dapat membantu untuk menyusun gambar atau ikon dalam *layout* lebih terstruktur dan rapi.
5. *Scale* (skala). Skala digunakan untuk mengukur ukuran elemen-elemen dalam *layout* agar memiliki ukuran yang sama, saling berkaitan, dan menarik.
6. *Alignment* (perataan), penggunaan *alignment* ini membantu untuk menyeimbangkan tampilan pada *layout* lebih terstruktur.
7. *Grid*, bentuk dari *grid* dapat berupa garis vertikal dan horizontal yang tidak terlihat yang ditempatkan pada interval yang pasti serta memberikan struktur untuk tata letak.
8. *Templates*, adalah kumpulan margin dan aturan penempatan yang diterapkan pada jenis konten dan *fitur* tertentu yang dapat menarik visual.

2.8 Kontrol

Kontrol merupakan bagian paling penting dalam pembuatan *user interface*. Menurut (Nauval et al., 2018) kontrol adalah segala sesuatu yang membuat pengguna dapat berinteraksi dengan ikon atau simbol melalui tampilan *Interface*. Menurut (Schlatter & Levinson, 2013) fungsi dari kontrol dalam aplikasi dibagi menjadi tiga kategori, yaitu:

1. *Navigation* (navigasi)

Seperti *tabs*, *scrollbars*, *dropdown menu* yang menunjukkan perpindahan dari satu tempat ke tempat lain dalam sebuah aplikasi.

2. *Data manipulation controls* (kontrol data manipulasi)

Seperti formulir dan *submission button* yang memberikan usulan cara untuk memilih dan mengelola informasi

3. *Information display controls* (kontrol tampilan informasi)

Seperti akordion dan lapisan yang mengusulkan untuk mengungkapkan informasi sesuai dengan permintaan.

2.9 Figma

Figma merupakan sebuah *software*, *tools*, atau aplikasi berbasis web yang digunakan untuk membuat desain UI (*User interface*) dan UX (*User Experience*). Figma dapat digunakan di sistem operasi *windows*, *linux* ataupun *macbook* yang terhubung ke internet. Kumpulan *fitur-fitur* yang ada didalam figma terfokus pada penggunaan UI/UX dengan penekanan kolaborasi *real time*. Selain memiliki *fitur* kelengkapan seperti Adobe XD, figma memiliki beberapa keunggulan yaitu dalam melakukan pekerjaan yang sama dapat dikerjakan lebih dari satu orang secara bersamaan meskipun ditempat yang berbeda. Dengan kemampuan aplikasi figma untuk melakukan pekerjaan secara kelompok hal ini membuat figma menjadi pilihan banyak bagi seorang designer UI/UX untuk membuat *Prototype website* atau aplikasi dalam waktu yang singkat.

2.10 Draw.io

Dalam membuat sebuah model desain *Interface* memerlukan sebuah alur atau arah bagaimana sebuah model UI jika akan di bentuk menjadi sebuah aplikasi jadi. Draw.io merupakan sebuah *platform* diagram online yang memungkinkan user atau pengguna membuat berbagai jenis diagram. Draw.io juga memungkinkan pengguna membuat

dengan cepat diagram alur, diagram organisasi, diagram UML, dan banyak tanpa perlu mengetahui secara detail tentang desain grafis.

Keunggulan dari *platform* draw.io adalah adanya *fitur* berbagai macam bentuk dan ikon bawaan, kemampuan dalam menyematkan diagram ke situs *web* atau dokumen, serta berkolaborasi tim secara waktu nyata.