

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Teori ini membantu dalam memahami bagaimana website dirancang dan diimplementasikan untuk mendukung *sistem pendaftaran Pelayanan KB di Klinik IBI Tegal*.

Penelitian sebelumnya berjudul *Implementasi Sistem Informasi Pelayanan Kesehatan Pada Klinik Smart Medica* menghasilkan kesimpulan bahwa pelayanan kepada masyarakat akan ditingkatkan juga sistem informasi Klinik Smart Medica bisa memberikan informasi terkini yang diperlukan pasien yang berujung pada kepuasan pasien. Sistem informasi ini dibangun menggunakan metode *Waterfall*, bahasa pemrograman yang dipakai *PHP* juga *Database MySQL* [6].

Penelitian yang dilakukan oleh Deni Ahmad Zakaria berjudul *Rancang Bangun Sistem Informasi Pendaftaran Akseptor Program Keluarga Berencana (KB) di DPPKBP2PA Kota Tasikmalaya*, beliau menyatakan bahwa sistem informasi pengolahan data akseptor KB (Keluarga Berencana) terus-menerus dikembangkan sesuai dengan kebutuhan yang terjadi baik ditingkat lokal, nasional maupun global. Aspek penting dari situasi ini yakni penyimpanan data secara manual. Sistem manual ini dapat menghambat kelancaran pengolahan juga pencairan data memiliki kesinambungan terhadap akseptor KB pada saat data dibutuhkan. Sebuah sistem informasi

suatu proses atau layanan dirancang guna mengatasi masalah tersebut, sistem ini memfasilitasi pengelolaan data dan mendukung kinerja yang efektif. Menurut hasil analisis juga perancangan sistem *informasi* mengenai pengolahan data, diharapkan dengan adanya sistem khusus pengolahan data mengenai akseptor KB ini diharapkan bisa mempermudah pengolahan datanya. Menurut metode tersebut, *informasi* jumlah akseptor di setiap kecamatan akan lebih mudah diakses. Sistem ini mampu melaporkan persentase keluarga yang memiliki akses terhadap capaian KB pada tiap kecamatan [1].

Menurut jurnal Sistem Informasi Pelayanan Pendaftaran dan Rekam Medis Di Klinik Chairna Medistra mengalami kendala dalam pengelolaan data pasien juga rekam medis cara manual, hal ini bisa menimbulkan ketidakefisienan juga potensi kekeliruan. Guna menangani permasalahan tersebut, penelitian ini mengusulkan pemanfaatan sistem *informasi* berbasis web yang menyatu guna memfasilitasi pendaftaran pasien juga pencatatan rekam medis. Temuan penelitian menyatakan bahwa penerapan *sistem informasi berbasis web di Klinik Chairna Medistra* menambah efisiensi pemberian layanan, akurasi data, juga kemudahan akses *informasi* pasien [2].

Beberapa penelitian terdahulu yang membahas mengenai *sistem informasi pelayanan pasien berbasis website pada puskesmas II Baturraden* menjadi bahan referensi peneliti yang dilakukan oleh Joko Dwi Mulyanto. Penelitian ini membuat sebuah aplikasi yang dibangun dalam bahasa pemrograman *php, html, javascript*. Adapun tujuan dari dibangunnya sistem

yakni guna membantu juga memudahkan proses pelayanan pasien yang memerlukan pengobatan, seperti membuat pencarian informasi pasien menjadi lebih cepat dan akurat [4].

Permasalahan pada bidan yakni kurangnya pengingat akan jadwal pemeriksaan ulang, hal ini menyebabkan banyak pasien KB yang lupa memeriksakan kembali keadaannya yakni menurut penelitian dari jurnal *Sistem Informasi Pelayanan KB Berbasis Web dengan Whatsapp Gateway*. Oleh karena itu, bidan masih menggunakan cara manual, dengan menuliskan kembali daftar pemeriksaan pada buku kohort pasien, hal ini akan memudahkan perencanaan pelayanan KB yang dilakukan oleh bidan setempat. Pada penelitian ini dibuat sistem informasi Pelayanan KB berbasis web dengan menggunakan aplikasi *Whatsapp*, sistem ini dimaksudkan guna memudahkan administrasi pelayanan KB kepada pasien dan mengingat jadwal pemeriksaan ulang [7].

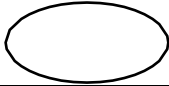
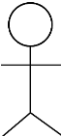




2.2 Landasan Teori

Bahasa visual guna memodelkan, mendokumentasikan, juga merancang sistem software yakni pengertian dari *UML (Unified Modeling Language)*. *UML* digunakan guna menggambarkan struktur, perilaku, interaksi, dan arsitektur dari sistem software yang akan dibangun [8]. Pada konteks perancangan website penjualan *voucher game*, penggunaan *UML* akan membantu dalam menggambarkan berbagai aspek dari sistem tersebut. Ini termasuk:

1. Diagram *Use Case*

Digunakan guna menggambarkan fungsionalitas sistem dari sudut pandang pengguna. Ini akan mencakup interaksi antara pengguna sistem misalnya, (pelanggan, admin) dan sistem pemesanan [9].






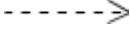

Tabel 2.1 Simbol *Use Case*

No	Simbol	Keterangan
1		<i>Use Case</i> : Abstraksi maupun interaksi antarsistem juga aktor.
2		<i>Actor</i> : Mewakili peran orang, sistem yang lain, maupun alat pada saat komunikasi dengan <i>use case</i> .
3		Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung juga bukannya mengindikasikan data.
4		Asosiasi antara aktor juga <i>use case</i> yang menggunakan panah terbuka guna mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5	 . <<include>>	<i>Include</i> , Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i> .
6	 <<extends>>	<i>Extend</i> , yakni perluasan dari <i>use case</i> lain jika kondisi maupun syarat terpenuhi.

2. Diagram *Class*

Digunakan untuk menggambarkan struktur statis dari sistem, termasuk kelas-kelas yang ada, atribut-atributnya, serta hubungan antara kelas-kelas tersebut misalnya, (hubungan pewarisan maupun asosiasi) [10].

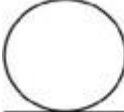
Tabel 2.2 Simbol Class

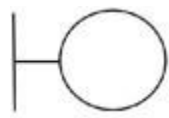

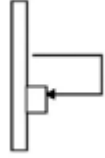


No	Gambar	Nama	Keterangan
1		<i>Generation</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku juga struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Usaha guna menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek- objek yang berbagi atribut serta operasi yang memadai.
4		<i>Collaboration</i>	Uraian dari rangkaian aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu <i>actor</i> .
5		<i>Realization</i>	Operasi yang benar-benar dilakukan suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

3. Diagram *Sequence*

Digunakan untuk menggambarkan urutan pesan maupun panggilan yang terjadi di antara objek-objek dalam sistem, menyoroti interaksi antara objek-objek tersebut dalam rangka mencapai satu tujuan [11].

Tabel 2.3 Simbol *Sequence*




No	Gambar	Nama	Keterangan
1		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data

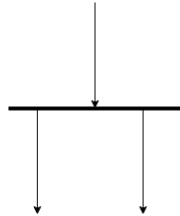
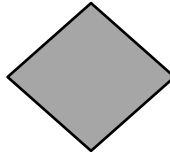
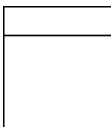
No	Gambar	Nama	Keterangan
2		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem
3		<i>Control Class</i>	Bertanggung jawab terhadap kelas – kelas terhadap objek yang berisi logika
4		<i>Recursive</i>	Pesan untuk dirinya
5		<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi
6		<i>Life Line</i>	Komponen yang digambarkan garis putus terhubung dengan objek

4. Diagram *Activity*

Digunakan untuk menggambarkan struktur fisik dari sistem, termasuk komponen-komponen perangkat lunak yang digunakan, serta cara komponen-komponen tersebut berinteraksi satu sama lain [12].

Tabel 2.4 Simbol *Activity*

No	Simbol	Keterangan
1		<i>Start Point</i> , diletakkan pada pojok kiri atas merupakan awal aktivitas.
2		<i>End Point</i> , akhir aktivitas.
3		<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.

No	Simbol	Keterangan
4		<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau guna menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Decision Points</i> , menggambarkan pilihan guna pengambilan keputusan, <i>true</i> atau <i>false</i> .
6		<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Adapun untuk tool yang digunakan dalam pembuatan aplikasi/ sistem pada penelitian ini adalah sebagai Berikut:

1. Visual Studio Code

Visual Studio Code (VS Code) yakni editor teks sederhana juga ringan yang dibuat oleh *Microsoft* guna sistem operasi berbagai platform. Tersedia untuk versi *Linux*, *Mac* juga *Windows*. Editor teks ini langsung kompatibel dengan bahasa pemrograman lain, hal ini dilakukan melalui penggunaan plugin yang dapat diinstal melalui *marketplace VS Code* seperti (*C++*, *C#*, *Python*, *Go*, *Java* juga beberapa bahasa pemrograman lainnya) [13].

Banyak fitur *VS Code* yakni *Intellisense*, *Git Intergation*, *Debugging* juga fitur tambahan yang menambah kemampuan editor teks. Fitur-fitur ini akan terus bertambah seiring bertambahnya versi *VS Code*,

ini juga diperbarui setiap bulan, dan inilah yang membedakan *VS Code* dengan editor teks lainnya.

Selain itu, editor teks *VS Code* bersifat *open source*, juga kodenya dapat diamati juga dimasukkan ke dalam pembuatannya. Kode sumber lengkap guna *VS Code* juga dapat ditemukan di link *Github*. Hal ini pula yang membuat *VS Code* populer di kalangan developer, sebab developer dapat ikut serta dalam pengembangan aplikasi menggunakan proses *VS Code* di kemudian hari.

2. HTML

HTML (Hypertext Markup Language) yakni metode menyediakan indikator yang menginstruksikan browser tentang struktur teks. *HTML* menginstruksikan *browser* tentang cara yang tepat untuk menyusun dokumen, *heading-nya* harus diatur, paragraf harus diatur, bagaimana teks akan muncul juga seterusnya. Informasi disediakan menunjukkan bahwa *browser* memiliki pengetahuan dasar tentang cara menampilkan berbagai komponen yang ada [14].

3. Bootstrap

Bootstrap yakni kerangka kerja *CSS* sumber terbuka juga bebas guna merancang situs *web* maupun aplikasi *web*. Kerangka kerja ini terdiri dari templat desain berbasis *HTML* juga *CSS* guna penulisan huruf, *formulir*, tombol, navigasi, juga komponen antarmuka lainnya, serta *JavaScript* yang disertakan secara opsional. Berbeda dengan *web* lain yang hanya fokus pada pengembangan *front-end* saja [15].

4. XAMPP

Pembuatan *XAMPP* berasal dari kumpulan pengembang yang disebut *Apache Friends*, yang terdiri dari Tim Inti (*Core Team*) [16], Tim Pengembang (*Development Team*) juga Tim Dukungan (*Support Team*). *Xampp* ialah kependekan dari kata yang artinya yakni: X=Program ini bisa dipakai pada berbagai sistem operasi seperti, *Windows, Linux, Mac Os*, juga *Solaris*. A=*Apache*, yakni aplikasi *WebServer*. Fungsi utama *Apache* ialah menghasilkan halaman web yang akurat bagi pengguna berdasarkan kode *PHP* yang digunakan oleh pembuat halaman web. Jika kode yang diperlukan ditulis dalam *PHP*, Database dapat diakses terlebih dahulu, misalnya (di *MySql*) guna mendukung website yang dihasilkan. M=*MySQL*, ialah aplikasi *database server*. Evolusinya disebut *SQL (Structured Query Language)*. *SQL* yakni bahasa terstruktur yang memfasilitasi pemrosesan database. *MySQL* digunakan untuk membuat juga memelihara database beserta isinya. Kita bisa memanfaatkan *MySQL* guna menambah, mengubah juga menghapus data dari database. P=*PHP*, bahasa pemrograman *website MySql*.

5. MYSQL

DMS yang *open source* yang memiliki dua lisensi berbeda, yakni *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas) ialah pengertian

dari *MySQL*. Maka, *MySQL* yakni *database server* yang gratis dilisensikan *GNU General Public License (GPL)*, lisensi ini tersedia untuk penggunaan pribadi atau komersial tanpa memerlukan lisensi yang sudah ada [17].

6. CSS

Pedoman yang digunakan guna menyusun beberapa komponen ke dalam sebuah *website* agar memiliki tampilan yang lebih terorganisir dan konsisten disebut *CSS (Cascading Style Sheets)*. Bahasa pemrograman tidak berhubungan dengan *CSS*. Sama halnya dengan *styles*, misalnya *heading*, *subbab*, *footer*, *images*, juga *style* lainnya bisa digabungkan menjadi beberapa file [18].

7. PHP

Rasmus Lerdorf ialah penemu pertama *PHP (Hypertext Preprocessor)* pada tahun 1995 yang merupakan seorang *Software Developer*. Rasmus saat itu bisa mengetahui jumlah pengunjung yang membaca *resume online*, inilah ide pertama *PHP*. Perancangan *script* memiliki dua tujuan, ialah menyimpan *informasi* tentang pengunjung situs web, kemudian menampilkan jumlah total pengunjung. Dan hingga saat ini, kedua tugas tersebut masih populer di dunia *web* saat ini. Setelah itu orang-orang di milis mulai berdiskusi tentang pembuatan *script* tersebut, akhirnya Rasmus Lerdorf merancang sebuah *tool / script* bernama *PHP* yang berasal dari *scripts* dalam bentuk *plaintext*. *PHP Interpreter* yakni aplikasi

perangkat lunak berbasis *Website* yang membaca file dan menafsirkannya, menyediakan output *HTML* juga memberikan instruksi tentang bagaimana perilaku yang ada maupun menginterpretasikan masukan dari pengguna [19].

Pengumpulan data dilakukan dengan menggunakan metode observasi dan metode pustaka dengan melakukan tahapan pengembangan sistem aplikasi dengan langkah – langkah seperti *Integration and Testing*.

1. *White Box Testing*

White box testing disebut sebagai pengujian structural. Yang mana perangkat lunak yang diuji tidak diketahui oleh penguji. Penguji didasarkan kepada spesifikasi kebutuhan dan tidak perlu dilakukannya analisis kode. Pengujian black box testing pengujian ini dilakukan dari sudut pandang pengguna akhir.

2. *Black Box Testing*

Pengujian black box testing disebut sebagai pengujian perilaku. Dimana struktur interior, logika perangkat lunak yang diuji tidak diketahui oleh penguji. Penguji didasarkan kepada spesifikasi kebutuhan dan tidak perlu dilakukannya analisis kode. Pengujian black box testing pengujian ini dilakukan dari sudut pandang pengguna akhir.