

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dasar teori dalam tugas akhir sistem arsip berbasis web meliputi konsep arsip dan manajemen dokumen elektronik, sistem arsip berbasis web, teknologi yang digunakan dalam pembangunan sistem arsip berbasis web, dan metode dan teknik pengembangan sistem arsip berbasis web[5]

Perkembangan pesat dalam teknologi informasi dan komunikasi saat ini membuat banyak orang menyadari pentingnya informasi. Media informasi dan telekomunikasi berfungsi sebagai sarana untuk melakukan transaksi informasi. Dengan kemajuan teknologi informasi dan telekomunikasi yang sangat cepat dalam dunia pengarsipan, perusahaan mulai memanfaatkan teknologi dalam proses kerja mereka[6].

Penelitian yang terdahulu dilakukan dengan judul “Perancangan *Website* Sebagai Media Promosi dan Informasi”. Permasalahan dalam penelitian ini terkait dengan promosi produk yang masih menggunakan metode yang kurang efisien. Oleh karena itu, dibutuhkan usaha untuk merancang sebuah situs web sebagai sarana untuk promosi dan penyampaian informasi, yang dapat mendukung pemasaran produk dengan cara yang lebih efektif dalam hal biaya, tenaga, dan waktu. Dengan demikian, menjadi lebih efisien dalam mencapai tujuan untuk memperluas bisnis. Dalam penelitian ini, perbedaannya tidak hanya terletak pada penyampaian informasi dan promosi,

tetapi juga lebih menggambarkan secara rinci produk oleh-oleh yang ditampilkan di dalam situs *website*[7].

Penelitian sebelumnya menunjukkan bahwa implementasi sistem arsip digital dapat secara signifikan mengurangi risiko kehilangan data dan meningkatkan efisiensi kerja. Martini (2021) dalam penelitiannya mencatat bahwa sistem arsip elektronik dapat memangkas waktu pencarian dokumen hingga 60%, yang memberikan efek baik bagi efisiensi organisasi. Di sisi lain, Isnawan dan Syaputra (2022) mengungkapkan bahwa penggunaan aplikasi arsip digital dinamis (SIMARDI) dalam pemerintah berhasil meningkatkan performa pegawai hingga 40%, karena data dapat diakses dengan lebih cepat dan jelas. Kerangka kerja[8].

Berdasarkan hal hal yang disebutkan di atas maka dari itu penulis melakukan penelitian terkait yang berjudul “ Perancangan Sistem Informasi Dokumen Berbasis Web pada Btpn Syariah Cabang Mms Tanjung” yang isinya tentang bagaimana sistem arsip dokumen yang beebasis web yang memfokuskan pada letak dokumen yang sudah tercantum di web dan memudahkn karyawan untuk menyimpan berkas yang sudah di input di website karena letak dokumen sudah tercantum di website. Menurut Isnawan dan Syaputra (2022) mengungkapkan bahwa penggunaan aplikasi digital dinamis(SIMARDI) dalam pemerintahan berhasil meningkatkan performa pegawai hingga 40% , karena data dapat di akses dengan cara cepat dan jelas.

2.2 Landasan Teori

2.2.1 Sistem informasi berbasis web

Adalah aplikasi yang memanfaatkan teknologi web untuk mengelola data dan informasi. Sistem ini dapat diakses melalui *browser web* di perangkat apa pun yang terhubung ke *internet*.

Ciri-ciri sistem informasi berbasis web

- a). Tidak perlu menginstal perangkat lunak tambahan
- b). Memberikan akses yang mudah dan cepat kepada pengguna
- c). Dapat diakses melalui *browser web*
- d). Memanfaatkan teknologi seperti *HTML*, *CSS*, dan *JavaScript*
- e). Menggunakan database untuk menyimpan data

2.2.2 *PHP (Hypertext Preprocessor)*

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan untuk membuat aplikasi web dinamis dan interaktif. *PHP* merupakan bahasa pemrograman *server-side* yang populer dan banyak digunakan, *PHP* adalah teknologi sumber terbuka yang dapat diakses secara gratis dan dikembangkan untuk memenuhi keperluan bisnis, pendidikan, atau profesional, yang didukung oleh komunitas pengguna dan pembuatnya. Untuk dapat menjalankan *PHP*, diperlukan sebuah *server web*[9].

2.2.3 *MySQL*

MySQL adalah sistem manajemen basis data relasional (*RDBMS*) *open-source* yang menggunakan bahasa kueri terstruktur (*SQL*) untuk

mengelola dan mengakses data dalam tabel yang terstruktur, Dimana *MySQL* mampu untuk melakukan banyak eksekusi perintah *query* dalam satu permintaan, baik itu menerima dan mengirimkan data. *MySQL* juga *multi-user* dalam arti dapat dipergunakan oleh banyak pengguna dalam waktu bersamaan. *MySQL* tersedia tersedia dalam perangkat lunak gratis dibawah lisensi *GNU General Public Lisenca (GPL)* dan juga menjual dalam lisensi komersial untuk keperluan jika pengguna tidak cocok menggunakan *lisensi General Public Lisenca (GPL)*. Pengguna *MySQL* yang merupakan sebuah *database server* sekaligus dapat sebagai *client*, dan dapat berjalan di *multi-OS (Operating System)* memiliki keunggulan lainnya seperti *Open Source* sehingga penggunaanya tidak perlu membayar lisensi kepada pembuatnya. Bahasa pemrograman yang dapat digunakan untuk mengakses *MySQL* diantaranya adalah dengan *C, C++, Java, Perl, PHP, Phyton*, dan *API*.

2.2.4 Sistem Arsip Berbasis Web Menurut Syarifuddin (2017),

Sistem arsip berbasis web adalah sistem yang memungkinkan pengguna untuk mengakses, mencari, dan mengelola dokumen dalam arsip menggunakan *internet*. Sistem ini dapat digunakan oleh organisasi atau individu untuk menyimpan dokumen elektronik dalam format digital dan mengelola dokumen tersebut. Teknologi yang Digunakan dalam Pembangunan Sistem Arsip Berbasis Web Menurut Suharjito (2019), teknologi yang digunakan dalam pembangunan sistem arsip berbasis web

meliputi *HTML (HyperText Markup Language)*, *CSS (Cascading Style Sheets)*, dan *JavaScript*[10].

2.2.5 Website

Situs web adalah sarana informasi yang tersedia di dunia maya. Tidak hanya berfungsi untuk menyebarkan informasi, *situs web* juga dapat digunakan untuk mendirikan toko *daring*. *Situs web* terdiri dari serangkaian halaman yang biasanya terorganisir dalam satu *domain* atau *subdomain*, yang posisinya berada di dalam *World Wide Web (WWW)* di Internet. Sebuah halaman situs adalah dokumen yang ditulis menggunakan format *HTML (Hyper Text Markup Language)*, yang hampir selalu bisa diakses melalui *HTTP*, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

2.2.6 .HTML (Hypertext Markup Language)

HTML adalah suatu bahasa kode yang dipakai untuk menciptakan halaman web, menampilkan berbagai informasi di dalam penjelajah web Internet dan format hiperteks sederhana yang ditulis dalam berkas format ASCII untuk menghasilkan tampilan yang terintegrasi. Dalam *HTML* digunakan tautan hiperteks atau keterkaitan antara teks dan dokumen lainnya. Oleh karena itu, pembaca dokumen dapat berpindah dari satu dokumen ke dokumen lainnya dengan mudah.

2.2.7 *Unified Modelling Language (UML)*

adalah sebuah "bahasa" yang telah diakui sebagai norma dalam bidang ini untuk visualisasi, perancangan, dan dokumentasi sistem perangkat lunak. *UML* memberikan sebuah format standar untuk merancang model suatu sistem. Dengan penerapan *UML*, kita dapat menciptakan model untuk berbagai tipe aplikasi perangkat lunak, yang dapat beroperasi di berbagai perangkat keras, sistem operasi, dan jaringan, serta dapat ditulis dengan menggunakan berbagai bahasa pemrograman. Namun, karena *UML* juga mengadopsi konsep kelas dan operasi, maka ia lebih cocok untuk pengembangan perangkat lunak menggunakan bahasa berbasis objek seperti *C++*, *Java*, *C#*, atau *VB.NET*. Meski demikian, *UML* masih dapat diterapkan untuk merancang aplikasi prosedural dalam *VB* atau *C*. Seperti pada bahasa pemrograman lainnya, *UML* menetapkan notasi serta sintaksis/semantik. Notasi *UML* terdiri dari sekumpulan simbol yang spesifik untuk merepresentasikan berbagai diagram perangkat lunak. Setiap simbol memiliki arti tersendiri, dan sintaksis *UML* menjelaskan cara kombinasi simbol-simbol tersebut. Notasi *UML* terutama berasal dari tiga notasi yang telah ada sebelumnya: *Grady Booch OOD (Desain Berorientasi Objek)*, *Jim Rumbaugh OMT (Teknik Pemodelan Objek)*, dan *Ivar Jacobson OOSE. (Object-Oriented Software Engineering)*[11]s.

Ada beberapa diagram yang ada di *UML*, yaitu:

a. *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* menggambarkan interaksi antara entitas pengguna dan sistem. *Use case* adalah aktivitas spesifik, seperti masuk ke dalam sistem, membuat daftar belanja, dan lain-lain. Seorang aktor adalah individu atau perangkat yang berinteraksi dengan sistem untuk menjalankan aktivitas tertentu. *Use case diagram* sangat bermanfaat ketika kita sedang merumuskan kebutuhan suatu sistem, berkomunikasi tentang desain dengan klien, dan membuat *test case* untuk semua fitur yang tersedia dalam sistem. Sebuah *use case* dapat menyertakan fungsionalitas dari *use case* lain sebagai bagian dari prosesnya. Umumnya, diasumsikan bahwa *use case* yang disertakan akan dipanggil setiap kali *use case* yang menyertakan tersebut dijalankan secara normal. Satu *use case* bisa disertakan oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik fungsionalitas yang umum. Selain itu, sebuah *use case* juga bisa memperpanjang *use case* lain dengan perilakunya sendiri. Sementara itu, hubungan generalisasi antara *use case* menunjukkan bahwa sebuah *use case* adalah spesialisasi dari *use case* yang lain[12].

b. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir

berawal, *decision* yang mungkin terjadi, dan bagaimana merekam 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 [13].

c. *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. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

d. 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 :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- 1) Private, tidak dapat dipanggil dari luar class yang bersangkutan
- 2) Protected, hanya dapat dipanggil oleh class yang

bersangkutan dan anak-anak yang Mewarisinya

3) Public, dapat dipanggil oleh siapa saja[14].

2.2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan langkah untuk mengidentifikasi kesalahan pada setiap elemen dari perangkat lunak, mendokumentasikan hasilnya, menilai semua aspek dari setiap bagian (sistem) dan menilai fitur-fitur dari perangkat lunak yang akan dibuat. Glen Myers mengemukakan beberapa pedoman yang bisa digunakan sebagai penjelasan mengenai pengujian perangkat lunak :

1. Pengujian merupakan sebuah proses eksekusi program dengan tujuan utama untuk mencari kesalahan
2. Sebuah kasus pengujian dikatakan baik jika memiliki kemungkinan penemuan kesalahan yang tinggi
3. Pengujian yang berhasil adalah pengujian yang menemukan kesalahan Pengujian

Berdasarkan ketiga pernyataan diatas dapat disimpulkan bahwa pengujian yang baik tidak hanya ditujukan untuk menemukan kesalahan pada perangkat lunak tetapi juga untuk dapat ditemukannya data uji yang dapat menemukan kesalahan secara lebih teliti dan cepat. Gambaran abstrak pengujian perangkat lunak dapat diilustrasikan program executable P (digambarkan dengan disket) dengan himpunan data uji T, dieksekusi pada komputer. Hasil dari pelaksanaan tersebut kemudian dibandingkan dengan

keinginan yang telah ditentukan dan dinilai. Fokus utama dari pengujian dalam situasi ini adalah bagaimana menghasilkan data uji T yang dapat menemukan kesalahan (fault) dalam program secara efisien, memberikan informasi mengenai kualitas program, dan memenuhi syarat atau kriteria yang ditentukan. Pengujian perangkat lunak mencakup berbagai aspek, yaitu pengujian yang dilakukan dengan metode atau teknik tertentu serta pengujian berdasarkan strategi yang diterapkan.[12], Metode pengujian merujuk pada pendekatan atau teknik yang digunakan untuk menguji perangkat lunak. Metode ini berkaitan dengan penyusunan data uji yang akan dijalankan pada perangkat lunak yang telah dibuat. Diharapkan metode pengujian memiliki sistem untuk menentukan data uji yang mampu melakukan pengujian secara menyeluruh dan memiliki peluang besar untuk menemukan kesalahan. Perangkat lunak dapat diuji dengan dua cara, yaitu :

- a. Pengujian dikerjakan dengan menjalankan data uji dan memeriksa apakah fitur perangkat lunak berfungsi dengan baik. Data uji dihasilkan dari spesifikasi perangkat lunak, yang dalam konteks ini menjelaskan fitur dari perangkat lunak tersebut. Metode pengujian ini dikenal sebagai Black Box Testing.
- b. Pengujian dilakukan dengan menggunakan data uji untuk memeriksa semua komponen dalam perangkat lunak (data

internal, perulangan, logika keputusan, jalur). Data uji dihasilkan dengan memahami struktur internal (kode sumber) perangkat lunak tersebut. Metode pengujian ini dikenal sebagai White Box Testing.

Metode pengujian white-box testing dan black-box testing memiliki berbagai jenis serta kriterianya. Penjelasan yang lebih mendetail tentang metode ini pengujian ini dapat dilihat pada lampiran[12]