

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini mengambil beberapa penelitian yang pernah dilakukan sebelumnya sebagai acuan dan pedoman dalam membangun aplikasi. Berikut adalah beberapa sumber acuan tersebut.

Tabel 2.1 Daftar Pustaka

Nama Peneliti	Topik	Metode	Teknologi	Interfce	Perbedaan Yang Sedang Diteliti
Agus Kurniawan (Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia), 2018	<i>Digital Rights Managemen t</i> Sebagai Solusi Keamanan Dokumen Elektronik	<i>Digital Rights Management</i>	RESTful API berbasis HTTP, HTML, PHP	WEB Service	Lebih menitik beratkan mekanisme DRM, tidak membangun sebuah sistem Front End.
Delberth J. Beti dan Yos Richard Beeh, S.T., M.Cs. (Program Studi Teknik Informatika,	Perancangan dan Implementasi <i>E-Library</i> Menggunakan <i>Digital Rights</i>		Android, RESTful API berbasis HTTP	Android	Platform menggunakan android native, aplikasi diterapkan secara

Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga), 2016	<i>Managemen t System</i> pada Android OS				online, proses menitikberatkan peminjaman buku transaksional
Muhammad Tedy Irwansyah (STMIK AKAKOM , Teknik Informatika), 2016	Teknologi Framework untuk membangun Model <i>E-Library</i> Pada Perpustakaan SMA BOPKRI Banguntapan		CodeIgniter	WEB	Berbasis WEB, tidak ada mekanisme DRM.
Alam, Universitas Indraprasta PGRI, 2016)	Kajian Efektivitas Penerapan <i>E-Library</i> : Studi Kasus Perpustakaan Nasional RI (Structural Equation	SEM			Hanya berisi metodologi pengukuran Tingkat <i>efektifitas e</i> – library yang sudah ada yaitu Perpustakaan

	Model (SEM))				n Nasional RI menggunakan beberapa metode pengujian.
Yoso Adi Setyoko (Fakultas Informatika, Institut Teknologi Telkom Purwokerto, 2020)	Desain Digital Right Management (DRM) Untuk Perangkat Lunak Berbasis Desktop Menggunakan Teknik Mutual Authentication	<i>Matual Authentication</i>			Melakukan perlindungan terhadap banyak perangkat lunak dengan menggunakan enkripsi yang dibangun menggunakan teori DRM

2.2 Dasar Teori

2.2.1 *E – Library*

E – library atau perpustakaan digital merupakan perpustakaan dimana keseluruhan koleksi yang dimiliki berformat digital yang dapat diakses oleh komputer ataupun media seluler, jenis pengaksesan *e – library* terdiri dari 2 (dua) cara melalui media *on – line* ataupun media *off – line*.

2.2.2 *Digital Right Management*.

Digital Right Management (DRM) system adalah istilah yang digunakan untuk mengatur data digital dan memproteksinya dari user yang tidak mempunyai hak akses. Data yang diamankan dapat berasal dari banyak bentuk antara lain Dokumen, Gambar, Musik, Video, dan lain sebagainya. (Agus Kurniawan, 2018).

Fungsi *digital right management* antara lain :

1. Perlindungan Hak Cipta

Fungsi utama DRM adalah melindungi hak cipta pemilik konten digital. DRM menerapkan mekanisme pengamanan yang kompleks untuk mencegah pembajakan, penyalinan ilegal, dan distribusi tidak sah dari konten digital.

2. Kontrol Penggunaan Konten

DRM memungkinkan pemilik hak cipta untuk mengontrol penggunaan konten digital mereka. Misalnya, DRM dapat membatasi jumlah perangkat yang dapat mengakses konten, membatasi waktu akses, atau membatasi kemampuan untuk menyalin atau mencetak konten.

3. Manajemen Lisensi

DRM juga berfungsi sebagai alat untuk mengelola lisensi penggunaan konten digital. Pemilik hak cipta dapat menetapkan syarat dan ketentuan dalam lisensi yang mengatur cara penggunaan konten, termasuk pembatasan geografis, jumlah pemutaran, atau durasi penggunaan.

4. Melindungi Karya Kreatif

DRM juga bertujuan untuk mendorong para pencipta karya digital dengan memberikan perlindungan hukum dan insentif ekonomi. Dengan adanya perlindungan hak cipta yang efektif, pencipta dapat merasa aman untuk mempublikasikan karya mereka dan memperoleh penghasilan yang adil dari hasil karya tersebut. (<https://sis.binus.ac.id/2023/05/15/digital-rights-management-di-balik-era-digitalisasi/>)

2.2.2.1 Key Diversification

Key diversification merupakan sebuah proses untuk melakukan identifikasi *private key* atau kunci privat. Dimana *private key* yang ditentukan akan mempengaruhi pada proses *Matual Autentication*.

2.2.2.2 Matual Autentication

Matual authentication merupakan autentikasi dua arah, dimana proses ini digunakan untuk melakukan otoritas tertentu, Setelah proses *key diversification* selesai maka proses selanjutnya adalah *mutual authentication*. Dimana *client* dan *server* masing-masing saling memberikan *challenge* dan *response*.

2.2.2.3 *Secure Channel*

Secure Channel merupakan pembuatan saluran pertukaran data yang aman antara ke dua objek atau dua *device* yang berbeda, pada proses ini dapat menggunakan beberapa mekanisme pengaksesan file.

2.2.3 **Flutter**

Flutter SDK merupakan perangkat *User Interface (UI) open-source* lintas *platform* yang dikembangkan oleh Google yang memungkinkan pengembang untuk mengembangkan aplikasi iOS, Android, Web (versi *beta*) dan Desktop (versi pengembangan) dengan basis kode yang sama. Sasaran pengembangannya adalah untuk memberikan aplikasi berkinerja tinggi di berbagai *platform* dan sangat cocok digunakan untuk mengembangkan aplikasi yang memerlukan tampilan yang indah. Fitur-fitur yang dimiliki oleh Flutter adalah :

1. *Hot reload* yang berjalan di *Dart Virtual Machine* untuk melihat perubahan tanpa melakukan *compile* ulang.
2. Setiap *project* flutter mempunyai sebuah *file* bernama *pubspec.yaml* atau sering juga disebut *pubspec*. *Pubspec file* menentukan *dependencies* yang dibutuhkan oleh sebuah *project* flutter seperti *package-package* tertentu (berserta versi-versinya), *font* dan juga gambar.
3. Kerangka kerja yang reaktif.
4. Kaya akan *widget* yang menerapkan *material design*.
5. *Dart devtools* untuk menguji dan men-*debug* aplikasi.

Flutter menggunakan bahasa pemrograman Dart yang bisa mudah dipahami jika sudah mengenal konsep pemrograman berorientasi objek (kelas,

metode dan variabel) dan konsep pemrograman imperatif (perulangan dan kondisional). (docs.flutter.dev, 2020).

2.2.4 Laravel

Laravel adalah *framework* aplikasi web kontemporer, *open-source* dan digunakan secara luas untuk perancangan aplikasi web yang cepat dan mudah. Laravel dibuat oleh Taylor Otwell dan berada dibawah lisensi MIT. *Framework* ini dibuat untuk pengembangan *website* mengikuti konsep MVC (*Model, View, Controller*). Konsep MVC merupakan suatu konsep yang memisahkan pengembangan aplikasi berdasarkan komponen- komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface* dan cara memproses datanya. Komponen pola MVC terdiri dari :

1. *Model* mewakili struktur data pada sistem aplikasi. Model berisi fungsi-fungsi yang membantu dalam memanipulasi data di dalam basis data (*insert, update, delete, search*).
2. *View* adalah bagian yang mengatur tampilan halaman *web* atau *user interface*.
3. *Controller* merupakan bagian yang menghubungkan model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

Laravel mempunyai sintaks yang ekspresif, jelas dan menghemat waktu. *Framework* ini dikembangkan dengan tujuan bahwa pengembangan web harus dapat dinikmati dan penuh kreatifitas. Pengembangan web dengan Laravel

mempermudah proses pengembangan web dengan mempermudah tugas-tugas yang umum seperti *routing, authentication, sessions, dan caching*. (B. Hermanto, 2019).

2.2.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah perangkat lunak yang merupakan sebuah implementasi dari sistem manajemen basis data relasional (RDBMS - *Relational Database Management System*) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan turunan yang bersifat komersial.

Pada berbagai kepentingan, MySQL memiliki fungsi dan kegunaan yang sangat penting dalam menangani data. Sebagai contoh, MySQL dapat menangani persediaan barang dan mencatat data penjualan dalam sistem toko *online (e-commerce)*. Dengan adanya sistem MySQL maka manajemen toko *online* dapat mengetahui persediaan barang yang ada didalamnya apakah masih tersedia atau kosong. Selain itu MySQL dapat memberikan laporan berupa informasi penjualan baik secara harian, mingguan, bulanan bahkan sampai tahunan. (YM Kusuma Ardhana., S.T., M.Kom., 2014).

2.2.6 Metode *Waterfall*

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Metode *waterfall* memiliki tahapan-tahapan sebagai berikut :

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

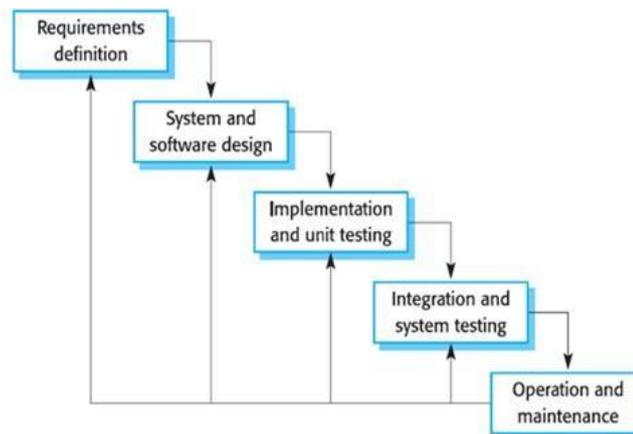
4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*.

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

Gambar 2.1 Bagan Model *Waterfall*



Gambar 2.1 adalah bagan metode *waterfall* yang merupakan metode pengembangan sistem yang digunakan pada penelitian ini. (Ginanjari Wiro Sasmito, 2017).

2.2.7 User Testing

User testing digunakan untuk memastikan apakah aplikasi dapat digunakan oleh pengguna dan sistem berjalan sesuai dengan rencana pengembangan.

2.2.8 Functional Testing

Functional testing meruakan proses pengujian yang akan digunakan, untuk memastikan bahwa aplikasi berjalan sesuai dengan rancangan yang telah dibuat.

2.2.8.1 Black - Box Testing

Pada *Black-box Testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Blackbox Testing* ini lebih menguji ke tampilan luar (*interface*) dari suatu aplikasi agar mudah digunakan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box Testing* bekerja dengan mengabaikan struktur *control* sehingga perhatiannya hanya terfokus pada informasi domain. (B. Hermanto, 2019).

Beberapa keuntungan penggunaan metode *Black-box Testing* antara lain adalah :

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.

2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. *Programmer* dan *tester* keduanya saling bergantung satu sama lain.

Berikut ini beberapa kekurangan dari metode *Black-box Testing* antara lain adalah :

1. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*.
2. Beberapa bagian *back-end* tidak diuji sama sekali. (Tri Snadhika Jaya, 2018).

2.2.8.2 White - Box Testing

White-box Testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan melihat modul untuk memeriksa dan menganalisis tentang kesalahan kode *program*. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan, singkatnya *White-box Testing* ini menguji dengan cara melihat *source code* dari suatu aplikasi (*software*) yang diuji tanpa memperdulikan tampilan antar muka atau UI (*User Interface*) dari aplikasi tersebut. (B. Hermanto, 2019).

White-box Testing merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Ketika program diterjemahkan ke dalam kode sumber bahasa pemrograman, maka akan terjadi kesalahan pengetikan. Banyak yang terdeteksi dengan mekanisme pemeriksaan sintaks, tetapi banyak juga yang tidak terdeteksi sampai dengan dimulainya uji coba. Karena alasan tersebut, maka uji coba *Whitebox Testing* diperlukan selain *Black-box Testing*. (Muhammad Nuris, 2015)