Sistem Informasi Perpustakaan Berbasis *Website* Dengan *Email Reminder*

Rino Ramadan [1,*], Andreyestha [2], Rahdian Kusuma Atmaja [3]

[1, 2, 3] Sistem Informasi; Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Kwitang, Kec. Senen, Kota Jakarta Pusat, DKI Jakarta, 10420, (021)54376399; e-mail: rino.rim@bsi.ac.id, andreyes2505@bsi.ac.id, rahdian.kusuma@bsi.ac.id.

* Korespondensi: e-mail: rino.rim@bsi.ac.id

Diterima:13 April 2021; Review: 14 Juni 2021; Disetujui: 22 Juni 2021

Cara sitasi: Ramadan R. Andreyesta, Atmaja RK. 2021. Sistem Informasi Perpustakaan Berbasis Website Dengan Email Remainder. *Information Management for Educators and Professionals*. Vol 5 (2): 11-20.

Abstrak: Perpustakaan pada umumnya merupakan tempat yang memiliki fungsi sebagai pusat sumber belajar dan sumber informasi bagi para pemakainya. Dalam menjalankan tugasnya sebagai seorang staf perpustakaan, mereka biasanya memiliki berbagai masalah yang dihadapi seperti proses pengolahan data yang manual ataupun menginformasikan kepada anggota perpustakaan untuk mengembalikan buku yang telah melewati batas waktu peminjaman. Masalah inilah yang kini ada di perpustakaan sekolah. Untuk mengatasi masalah-masalah yang ada maka perlunya dibuat sebuah aplikasi perpustakaan dengan memanfaatkan teknologi informasi dengan dipadukan layanan internet dan *email*. Maka dari itu penulis mencoba membuat Penelitian mengenai sistem informasi perpustakaan dengan metode *waterfall model* serta berbasis *website* yang dilengkapi sistem *email reminder*. Sistem ini merupakan solusi yang terbaik guna mengatasi permasalahan-permasalahan yang ada pada perpustakaan sekolah, sehingga proses pengolahan data akan menjadi lebih mudah, cepat, tepat dan akurat. Selain itu, dengan memanfaatkan teknologi internet dan email lebih memudahkan petugas perpustakaan untuk memberikan informasi mengenai batas waktu peminjaman buku. Sistem ini membuat proses menjadi lebih efektif dan efisien dibandingkan dengan sebelumnya.

Kata kunci: aplikasi, pengingat *email*, perpustakaan, *website*

Abstract: *Libraries in general are a place that has a function as a learning resource center and a source of information for its users. In carrying out their duties as library staff, they usually have various problems, such as manual processing of data or informing library members to return books that have passed the borrowed time limit. This problem is now in the library of school. To solve the existing problems, it is necessary to make a library application by utilizing information technology combined with internet and email services. Therefore, the author tries to make research on library information systems with the waterfall model method and website-based program equipped with an email reminder system. This system is the best solution to solve the problems that exist in the school library, so that the data processing will be easier, faster, more precise and accurate. In addition, utilizing internet and email technology makes it easier for librarians to provide information about the deadline for borrowing books. This system makes the process more effective and efficient than before.*

Keywords*: apps, libraries, reminder emails, websites*

1. Pendahuluan

Perpustakaan pada umumnya adalah tempat yang berfungsi sebagai pusat sumber belajar dan sumber informasi bagi pemakainya, baik itu staf perpustakaan maupun para pembaca yang berkunjung ke perpustakaan tersebut.

Oleh karena itu, penting kiranya mengenal peran seorang staf perpustakaan dalam mengelola sebuah perpustakaan, apa yang harus dilakukan terhadap koleksi perpustakaan agar informasi yang terdapat didalamnya dapat bermanfaat bagi pembaca perpustakaan tersebut.

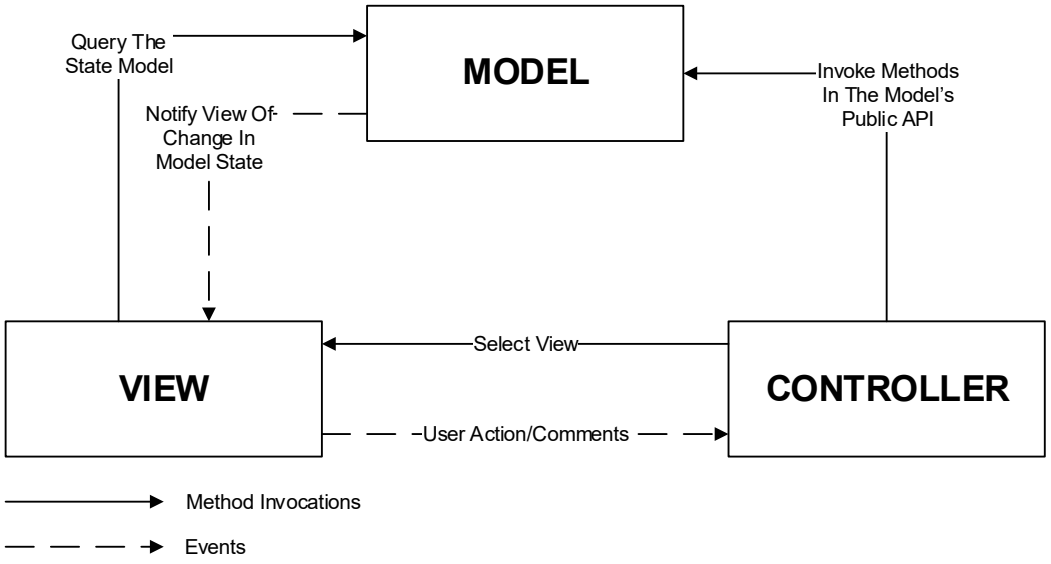
Dalam menjalankan tugasnya seorang staf perpustakaan biasanya memiliki masalah dengan harus mendata semua buku. Misalnya buku yang dipinjam, buku yang rusak, data siswa yang meminjam dan sebagainya. Hal itu bertujuan agar koleksi yang terdapat dalam perpustakaan tersebut dapat terjaga dengan baik. Buku yang menumpuk dan tidak tertata dengan rapih akan mengganggu kegiatan perpustakaan dan bisa menyebabkan tercecer atau bahkan hilang. Masalah inilah yang kini ada di perpustakaan sekolah. Masalah lainnya ialah sulitnya memberitahu kepada siswa bahwa buku yang dipinjam sudah lewat masa peminjaman.

Selain itu, untuk melengkapi hasil akreditasi sekolah yang diselenggarakan oleh tim Badan Akreditasi Nasional (BAN) di sekolah, maka segala aspek yang berada dalam ruang lingkup sekolah telah dinilai termasuk didalamnya perpustakaan. Untuk memenuhi saran kelengkapan tersebut serta untuk mengatasi masalah-masalah yang ada maka perlunya dibuat sebuah aplikasi perpustakaan dengan memanfaatkan teknologi informasi dengan dipadukan layanan internet dan *email*.

Dengan adanya masalah-masalah diatas, serta meninjau pentingnya aplikasi perpustakaan tersebut, maka dengan pembuatan sistem aplikasi perpustakaan akan dapat menyelesaikan masalah-masalah yang ada serta mempermudah siswa maupun staf perpustakaan dalam menjalankan tugasnya.

2. Metode Penelitian

Sistem informasi perpustakaan berbasis *website* ini dirancang dengan konsep arsitektur *MVC* atau *Model View Controller*. *Model*, *view* dan *controller* sangat erat hubuganya, oleh karena itu, mereka harus merujuk satu sama lain. Gulzar menjelaskan dalam penelitian Dayat dan Anggriani sebagaimana gambar 1 [1].



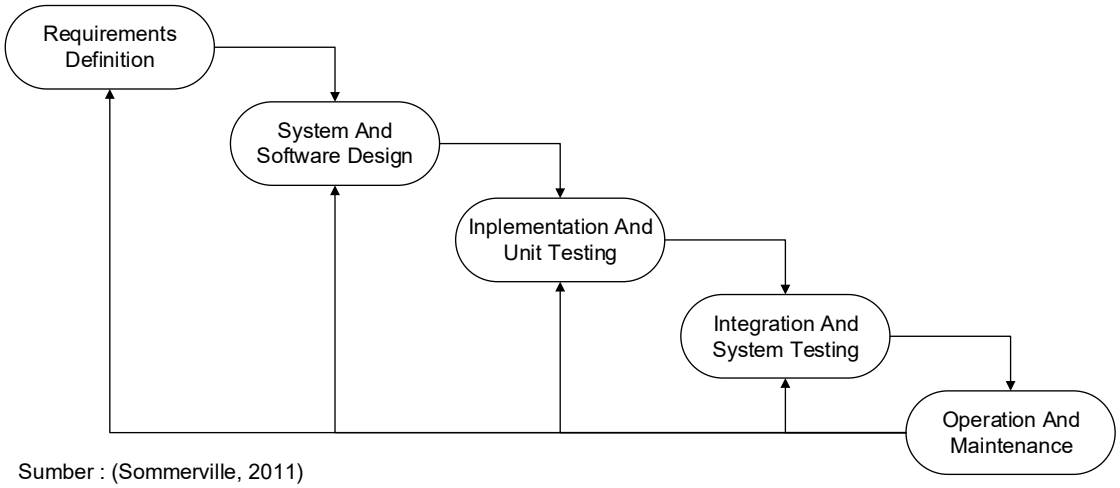
Sumber : (Gulzar, 2002)

Gambar 1. Hubungan antara *model*, *view*, dan *controller*.

Menurut Ramadan dalam penelitiannya, penerapan konsep *MVC* dalam pembangunan sistem informasi manajemen *data* terintegrasi bertujuan untuk memberikan kesederhanaan dan kemudahan bagi programmer *website* dalam pemeliharaan(*maintenance*) sistem, karena memisahkan data (*model*) dari segi tampilannya (*view*) dan cara bagaimana mengolahnya (*controller*), sehingga tidak menyulitkan saat harus memperbaiki atau mengembangkan sistem di masa yang akan datang [2].

Vol. 5, No. 2, Juni 2021, 11 - 20

Selanjutnya untuk metodologi pengembangan sistem menggunakan *SDLC* dengan model pengembangan sistem *waterfall*. *Waterfall* adalah sebuah proses dasar seperti spesifikasi, pengembangan, validasi, evolusi dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya [3], yang disajikan pada gambar 2.



Sumber : (Sommerville, 2011)

Gambar 2. Fase-fase dalam *Waterfall Model*.

Berikut ini penulis berikan penjelasan mengenai tahapan-tahapan waterfall menurut gambar diatas. Requirements analysis and definition merupakan proses pengumpulan kebutuhan secara komprehensif, lalu dilakukan analisa dan didefinisikan kebutuhan yang harus dipenuhi oleh program nanti. *System and software design* merupakan proses desain sistem yang dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap. *Implementation and unit testing* merupakan proses desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibuat kemudian dilakukan pengujian secara per modul. *Integration and system testing* merupakan penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*). *Operation and maintenance* didefinisikan sebagai proses operasional program aplikasi dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

3. Hasil dan Pembahasan

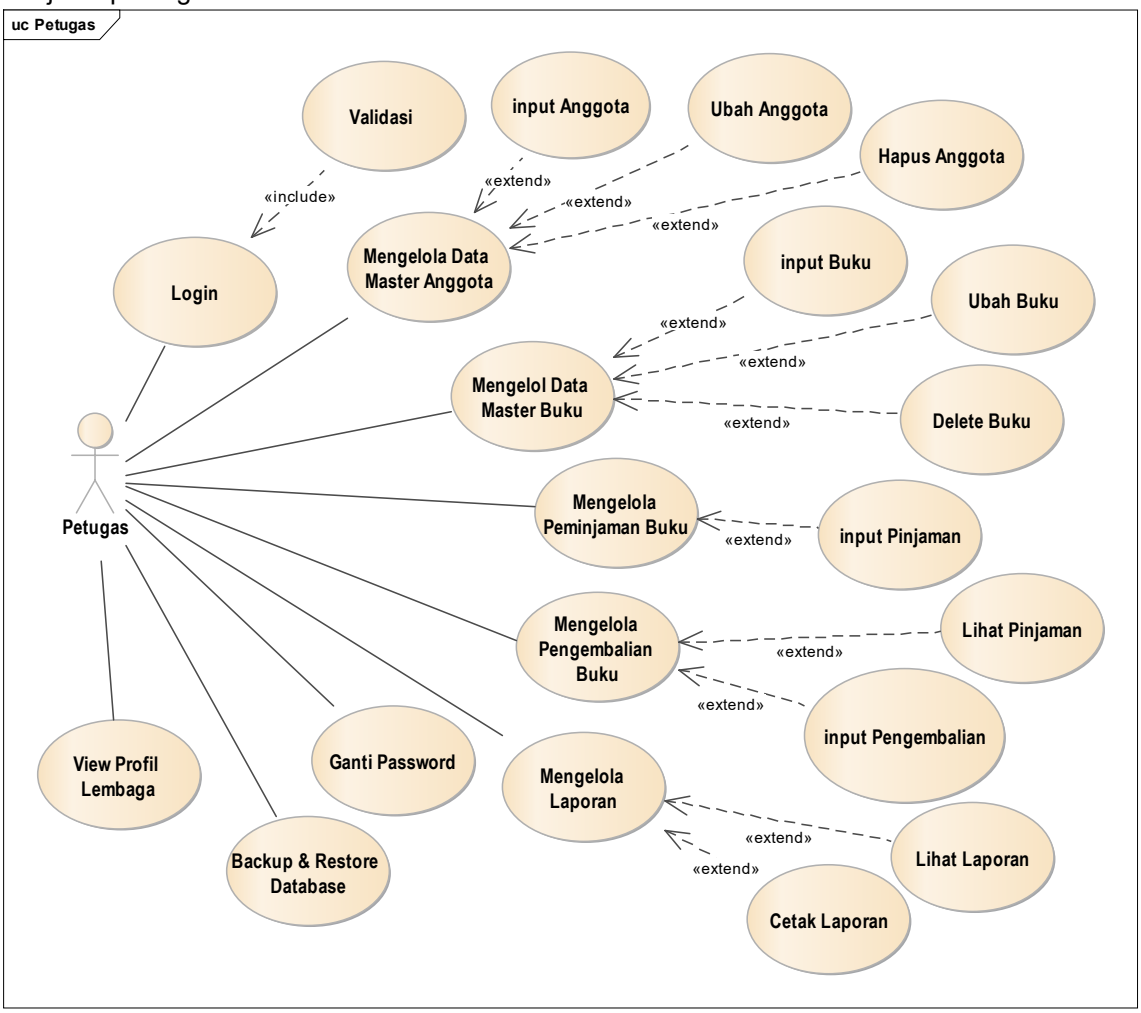
Berdasarkan hasil pengamatan pada proses yang terdapat di perpustakaan sekolah, terdapat beberapa kebutuhan pengguna dari sisi petugas, anggota dan administrator terhadap sistem informasi perpustakaan. Petugas dapat mengelola data master anggota, data master buku, transaksi pinjaman buku, transaksi pengembalian buku, melakukan pengaturan ganti *password*, mengelola *backup* dan *restore* program, melihat profil lembaga, melihat katalog buku dan mengelola laporan (kecuali laporan pengguna). Administrator dapat mengelola data master pengguna, data master anggota, data master buku, transaksi pinjaman buku, transaksi pengembalian buku, melakukan pengaturan ganti *password*, mengelola pengaturan (*setting*) program, *backup* dan *restore* program, melihat profil lembaga, melihat katalog buku, mengelola semua laporan. Anggota dapat melihat katalog buku.

Desain Sistem

Unified Modeling Language selanjutnya disebut (*UML*) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi sistem [4]. Dalam penelitian kali ini, penulis menggunakan 3 bahasa grafis pengembangan sistem dalam proses dokumentasi.

Pertama : *Use Case Diagram*, menurut Ade Hendini dalam penelitiannya, use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [5], Dalam hal ini disajikan pada gambar 3.



Gambar 3. *Use Case* Diagram Petugas.

Di *use case* diagram petugas, penulis mendeskripsikan kelakuan (*behavior*) dari petugas adalah sebagai berikut. Petugas melakukan *login* dengan syarat validasi terhadap login harus terpenuhi terlebih dahulu. Petugas mengelola data master anggota dengan kelakuan (*behavior*) tambahan berupa proses *input*, ubah dan hapus data anggota. Petugas mengelola data master buku dengan kelakuan (*behavior*) tambahan berupa proses *input*, ubah dan hapus data anggota. Petugas mengelola data peminjaman buku dengan kelakuan (*behavior*) tambahan berupa proses *input data* peminjaman buku. Petugas mengelola data pengembalian buku dengan kelakuan (*behavior*) tambahan berupa proses input pengembalian buku dan lihat data peminjaman buku. Petugas mengelola data laporan dengan kelakuan (*behavior*) tambahan berupa proses cetak dan lihat laporan. Petugas dapat mengganti *password*, *backup* & *restore database* dan melihat profil lembaga.

Kedua: *Activity* Diagram, menurut Rachman dalam penelitiannya, *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak [6]. *Activity* diagram yang penulis buat merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity* diagram yang penulis buat juga digunakan untuk mendefinisikan atau mengelompokan aluran tampilan dari sistem yang penulis buat. *Activity* diagram yang penulis buat memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah kepada aktivitas yang

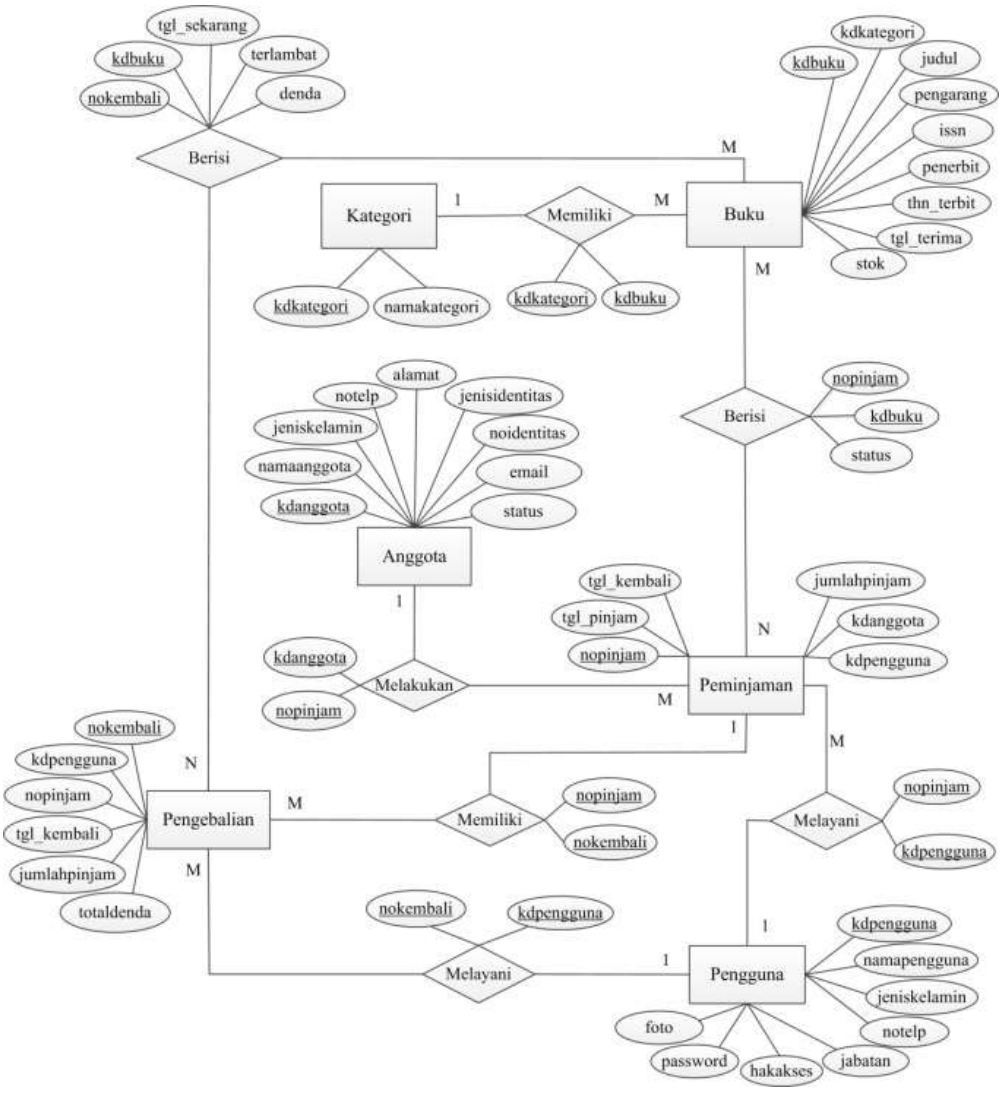
terjadi dari awal hingga akhir. *Activity* diagram tersebut terdiri dari penginputan *data* master, menjelaskan mengenai aktivitas aliran bisnis pemrosesan penginputan *data* master.
Pada *activity* diagram ini, administrator melakukan parameter *input*, simpan, ubah dan hapus terhadap data master kepada sistem. *Activity* diagram penginputan *data* peminjaman, menjelaskan mengenai aktivitas aliran bisnis pemrosesan penginputan data peminjaman. Pada *activity* diagram ini, administrator melakukan parameter *input* dan simpan terhadap data peminjaman kepada sistem.

Ketiga *Logical Record Structure*, menurut Sukmaindrayana dan Sidik dalam penelitiannya, *logical record structure* (*LRS*) digambarkan kotak persegi panjang dan dengan nama yang unik. *File record* pada *LRS* ditempatkan dalam kotak. *LRS* terdiri dari *link* atau hubungan antara tipe *record* lainnya, banyaknya *link* dari *LRS* yang diberi nama oleh filed-filed yang kelihatan pada kedua *link* tipe *record* [7]. Menurut Septiani, Afni dan Andharsaputri dalam penelitiannya, *logical record structure* (*LRS*) adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tabel dan *foreign key*. *Model* sistem yang digambarkan dengan sebuah *entity relationship* diagram akan mengikuti pola atau aturan pemodelan tertentu dalam kaitannya dengan konversi ke *LRS*, maka perubahan yang terjadi adalah mengikuti aturan sebagai berikut. Setiap entitas akan dirubah menjadi bentuk kotak. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada *entitiy relationship* diagram *1:M* (relasi bersatu dengan kardinalitas *M*) atau tingkat hubungan *1:1* (relasi bersatu dengan kardinalitas yang paling membutuhkan referensi), sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya *M:M (many to many)* dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan [8].

Menurut Riyanto dalam bukunya, beliau mendefinisikan, "*logical record structure* adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas". *Logical record structure* dibentuk menggunakan nomor dari tipe *record*. *Logical record structure* terdiri dari hubungan antar tipe *record*. *Link* tersebut menunjukan arah dari satu tipe *record* lainnya. Banyak link atau hubungan dari *LRS* yang diberi tanda field-field yang kelihatan pada kedua *link* tipe *record*. Penggambaran *LRS* mulai dengan menggunakan *model* yang dimengerti. Beberapa metode yang dapat digunakan, dimulai dengan hubungan kedua *model* yang dapat dikonversikan ke *LRS*. Metode yang lain dimulai dengan *ER-diagram* dan langsung dikonversikan ke *LRS* [9].

Keempat adalah: *Entity Relationship* Diagram, menurut Fathansyah dalam penelitian Sukmaindrayana dan Sidik menyimpulkan bahwa, *Entity Relationship Diagram* (*Diagram E-R*) adalah yang digunakan untuk menggambarkan *model Entity Relationship* yang berisi komponen-komponen. Himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasi seluruh fakta dari dunia nyata yang kita tinjau. Kegunaan *entity relationship* diagram adalah diagram yang digunakan untuk memodelkan struktur *data* dan hubungan antar *data*, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan. Simbol tersebut merupakan entitas, atribut, dan hubungan relasi. Entitas merupakan sesuatu yang mewakili objek yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entitas ini biasanya digambarkan menggunakan persegi panjang. Keberadaan entitas biasanya berdiri sendiri dan digambarkan (dipresentasikan) dengan sekumpulan atribut. atribut merupakan simbol yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut memiliki sesuatu yang dapat mengidentifikasikan isi dari sebuah elemen satu dengan elemen yang lain. Gambar atribut diwakili oleh simbol *elips*. Ada 2 jenis atribut *stored attribute* dan *derived attribute*. *Stored attribute* merupakan atribut yang langsung terlihat pada entitas (atribut nama, atribut alamat). *Derived attribute* merupakan atribut hasil perhitungan dari atribut yang lain (misal atribut umur dihitung dari atribut tanggal lahir). Hubungan dari relasi adalah hubungan yang terjadi antara

suatu himpunan dengan himpunan entitas yang lainnya. Pada penggambaran diagram hubungan entitas, relasi adalah perekat yang menghubungkan antara satu entitas dengan entitas lainnya. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu *database* yaitu *one to one, one to many* dan *many to many*. Satu ke satu *(One to one)* merupakan hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B. Satu ke banyak *(One to many)* merupakan hubungan relasi satu ke banyak yaitu setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan entitas pada himpunan entitas A. Banyak ke banyak *(Many to many)* merupakan hubungan relasi banyak ke banyak yaitu setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. [7].



Sumber: Hasil Penelitian (2021)

Gambar 4. *Entity Relationship Diagram*.

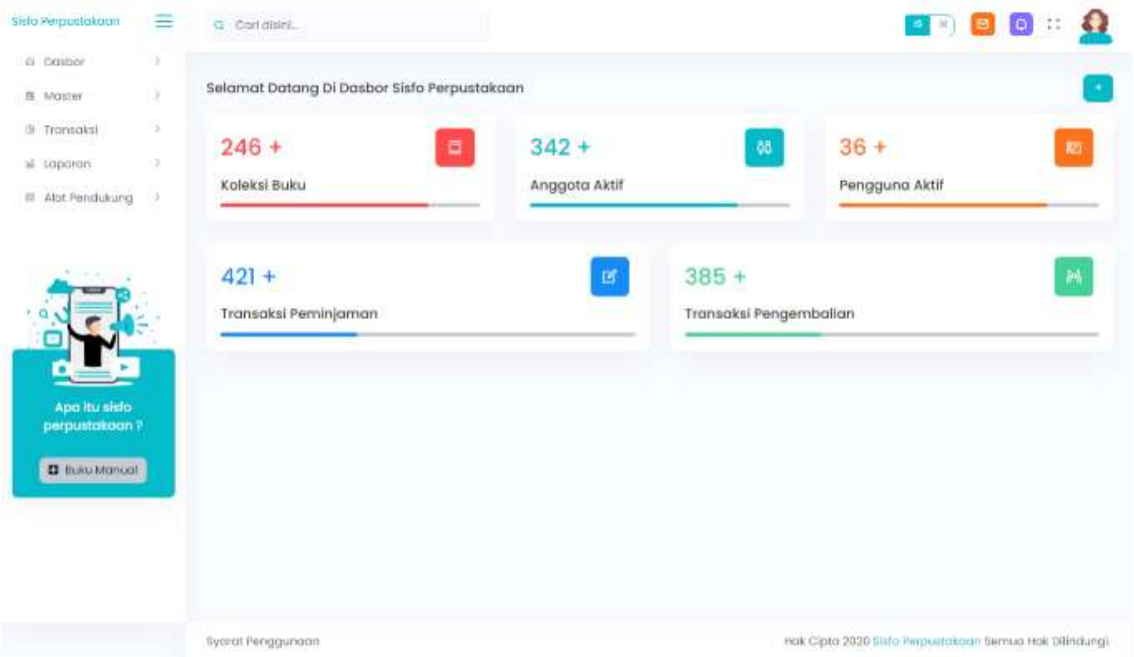
Di *entity relationship* diagram yang penulis buat. Penulis menggambarkan sebuah pemodelan *data* utama yang membantu mengorganisasikan data ke dalam entitas-entitas dan menentukan hubungan antar entitas dari setiap *data*. *Data* tersebut merupakan data

Vol. 5, No. 2, Juni 2021, 11 - 20

pengembalian, anggota, pengguna dan peminjaman. *Data* pengembalian terdiri dari nokembali, kdpengguna, nopinjam, tgl_kembali, jumlahpinjam dan total denda. *Data* pengembalian memiliki relasi terhadap *data* buku dengan kardinalitas *many to one*, terhadap data pengguna dengan kardinalitas *one to many* dan terhadap data peminjaman dengan kardinalitas *many to many*. *Data* pengguna terdiri dari foto, *password*, bukaakses, jabatan, notelp, jeniskelamin, namapengguna dan kdpengguna. *Data* pengguna memiliki relasi terhadap *data* pengembalian dengan kardinalitas one to many dan terhadap data peminjaman dengan kardinalitas one to many. *Data* peminjaman terdiri dari nopinjam, tgl_pinjam, tgl_kembali, jumlahpinjam, kdanggota dan kdpengguna. *Data* peminjaman memiliki relasi terhadap data pengguna dengan kardinalitas *one to many*, terhadap data pengembalian dengan kardinalitas *many to many*, terhadap *data* buku dengan kardinalitas *many to one* dan terhadap *data* buku dengan kardinalitas *many to many*. *Data* anggota terdiri dari kdanggota, namaanggota, jeniskelamin, notelp, alamat, jenisidentitas, noidentitas, *email* dan status. *Data* angota memiliki relasi terhadap data peminjaman dengan kardinalitas *one to many*. *Data* buku terdiri dari kdbuku, kdkategori, judul, pengarang, *issn*, penerbit, thn_terbit, tgl_terima, stok. *Data* buku memiliki relasi terhadap *data* peminjaman dengan kardinalitas *many to many*, terhadap *data* pengembalian dengan kardinalitas *many to many* dan terhadap *data* ketegori dengan kardinalitas *many to one*. *Data* kategori terdiri dari kdkategori dan namakategori. *Data* kategori memiliki relasi dengan *data* buku dengan kardinalitas *one to many*.

Implementasi Tampilan

Gambar 5 menunjukan tampilan antar muka aplikasi sistem informasi perpustakaan yang penulis buat pada penelitian kali ini.



Gambar 5. Tampilan halaman utama.

Tampilah halaman utama diatas merupakan tampilan yang menjadi halaman awal yang akan digunakan oleh petugas untuk melakukan proses pengelolaan *data* master anggota, *data* master pengguna, *data* master buku, *data* peminjaman dan *data* pengembalian. Ditampilan tersebut juga dijelaskan jumlah koleksi buku yang dimiliki oleh perpustakaan, jumlah anggota aktif yang terdapat di perpustakaan, jumlah anggota aktif yang terdapat di perpustakaan, jumlah transaksi peminjaman yang terjadi di perpustakaan dan jumlah transaksi pengembalian yang terjadi di perpustakaan.

Pengujian Sistem

Menurut Zweben dalam penelitian babaeian, tujuan dari pengujian perangkat lunak adalah untuk mengetahui apakah perangkat lunak tersebut benar atau tidak [10].

Tujuan dilakukannya pengujian sistem informasi perpustakaan berbasis *website* adalah untuk mengetahui kesesuaian antara modul sistem yang sudah di implementasikan dengan kebutuhan yang sudah di definisikan sebelumnya pada tahap analisa kebutuhan sistem. Pengujian dilakukan dengan beberapa *web browser* yang umum digunakan, untuk memeriksa setiap kesalahan baik dari proses *coding* (logika maupun sintaks) sampai keluaran program, sehingga dapat segera diperbaiki segala kekurangannya.

Pengujian sistem informasi perpustakaan berbasis *website* menggunakan metode *black box testing*.

Menurut Mustaqbal dalam penelitian Cholifah, *black box testing* pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode *black box testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari *data* yang di harapkan, estimasi banyaknya *data* uji dapat dihitung dari banyaknya *field data* masukan yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan *data* yang tidak diharapkan maka menyebabkan *data* yang disimpan kurang *valid* [11].

Menurut Ammann dalam penelitian jaya, keuntungan penggunaan metode *black box testing* adalah : (1) penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) pengujian tersebut 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. Kekurangan dari metode *Blackbox Testing* adalah : (1) uji kasus sulit di desain tanpa spesifikasi yang jelas; (2) kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*; (3) beberapa bagian *back end* tidak diuji sama sekali [12].

Pengujian *black box* dilakukan dengan cara memberi *input* dari pengguna kepada sistem yang sudah berjalan kemudian mengamati hasil *output* dari sistem. Pengujian dilakukan pada setiap *use case* untuk mengetahui kesesuaian fungsi sistem, dengan prosedur sebagai berikut: (1) Menentukan *data* yang akan digunakan untuk keperluan pengujian perangkat lunak, (2) Menentukan metode pengujian yang akan digunakan, (3) Melakukan pengujian pada setiap *use case* menggunakan *data* yang sudah disiapkan dan membandingkan hasilnya dengan kriteria pengujian, seperti dalam tabel 1 berikut ini:

Tabel 1. Daftar *Use Case* dan Kriteria Evaluasi Hasil Pengujian

***ID* Kasus Uji**	***Use Case***	**Kriteria Evaluasi Hasil**	**Pengujian Kesimpulan**
UC-1	Mengelola *data* anggota	Sistem dapat mengelola *data* anggota untuk semua *user* yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	Valid
UC-2	Mengelola *data* buku	Sistem dapat mengelola *data* buku untuk semua *user* yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	Valid
UC-3	Mengelola *data* pengguna	Sistem dapat mengelola *data* pengguna untuk semua *user* yang memiliki hak akses administrator sesuai dengan ketentuan	Valid
UC-4	Mengelola *data* transaksi	Sistem dapat mengelola *data* transaksi untuk semua *user* yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	Valid
UC-5	Mengelola *data* password	Sistem dapat mengelola data *password* untuk semua *user* yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	Valid
UC-6	Mengelola *data backup* dan *restore*	Sistem dapat mengelola *data backup* dan *restore* untuk semua *user* yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	Valid
UC-7	Mengelola *data* profil	Sistem dapat mengelola data profil untuk semua	Valid

ID Kasus Uji	*Use Case*	Kriteria Evaluasi Hasil	Pengujian Kesimpulan
		user yang memiliki hak akses petugas dan administrator sesuai dengan ketentuan	
UC-8	Mengelola *data* katalog	Sistem dapat mengelola *data* katalog untuk semua *user* sesuai dengan ketentuan	Valid
UC-9	Mengelola *data* laporan	Sistem dapat mengelola *data* laporan untuk semua *user* yang memiliki hak akses petugas(kecuali laporan pengguna) dan administrator sesuai dengan ketentuan	Valid

Sumber : Hasil Penelitian (2021)

4. Kesimpulan

Setelah melalui pembahasan yang mendalam, berikut ini adalah kesimpulan dari sistem informasi perpustakaan yang penulis dapatkan, antara lain. Sistem informasi ini berhasil meningkatkan efektifitas pendataan buku, sehingga semua stok buku dapat dipastikan dengan akurat. Dengan adanya penerapan sistem informasi perpustakaan maka proses peminjaman dan pengembalian buku berhasil lebih akurat dan terjamin, sehingga tidak ada lagi masalah buku tersebut hilang tidak diketahui sebabnya. Sistem informasi ini berhasil meningkatkan efektifitas kerja petugas perpustakaan, sehingga lebih memudahkan segala pengolahan datanya. Dengan memanfaatkan teknologi internet dan *email*, maka akan lebih mudah mengingatkan kepada anggota perpustakaan apabila telah melewati batas peminjaman buku.

Adapun untuk saran, disini penulis mencoba memberikan saran baik kepada pengguna aplikasi maupun kepada pembaca agar dalam pelaksanaannya dapat berjalan dengan baik, antara lain. Pentingnya *file* cadangan (*backup*) untuk menghindari kehilangan *data* yang tersimpan di komputer karena berbagai faktor penyebab, serta pada periode tertentu perlu adanya pengecekan kembali guna menjaga dan meningkatkan kualitas aplikasi tersebut. Perlu adanya pengetahuan dan pelatihan bagi petugas dalam menjalankan aplikasi yang digunakan. Diperlukan perawatan komputer secara berkala untuk menghindari rusaknya perangkat komputer. Bagi pembaca yang ingin melanjutkan penelitian ini, diharapkan agar membahas mengenai bagaimana proses pendataan jika buku yang dipinjam oleh anggota hilang. Selain itu, perlu dibahas lebih mendalam juga *supplier* buku yang berperan dalam pengadaan stok buku di perpustakaan sekolah.

Referensi

[1] A. R. Dayat and L. Angriani, “Pemanfaatan Model-View-Controller (MVC) Dalam Rancang Bangun Sistem Informasi Rakornas Aptikom 2017,” *Semin. Nas. APTIKOM*, no. November, pp. 416–420, 2017.
[2] R. Ramadan, “Penerapan Konsep Model View Controller Pada Sistem Informasi Manajemen Data Terintegrasi,” *Inf. Manag. Educ. Prof.*, vol. 5, no. 1, pp. 45–54, 2020.
[3] Sommerville, *Software Engineering (9th ed.)*. Massachusetts: Addison-Wesley, 2011.
[4] S. Mulyani, *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sistematika, 2016.
[5] A. Hendini, “Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang,” *J. Khatulistiwa Inform.*, vol. 2, no. 9, pp. 107–116, 2016, doi: 10.1017/CBO9781107415324.004.
[6] Rachman, “Sistem Informasi Wisata Di Ampera Waterpark,” *J. Siliwangi*, vol. 4, no. 2, pp. 87–92, 2018, [Online]. Available: http://jurnal.unsil.ac.id/index.php/jssainstek/article/download/570/369.
[7] A. Sukmaindrayana and R. Sidik, “Aplikasi Grosir Pada Toko RSIDIK Bungursari Tasimalaya,” *J. Manaj. Inform.*, vol. 4, no. 2, pp. 31–40, 2017, [Online]. Available: https://www.cambridge.org/core/product/identifier/CBO9781107415324A009/type/book_part.
[8] M. Septiani, N. Afni, and R. L. Andharsaputri, “Perancangan Sistem Informasi Penyewaan Alat Berat,” *JUSIM (Jurnal Sist. Inf. Musirawas)*, vol. 4, no. 02, pp. 127–135, 2019, doi: 10.32767/jusim.v4i02.639.
[9] Riyanto, *Migrasi Micosoft SQL server dengan Postgre SQL*. 2011.
[10] M. Babaeian, V. Ghasemiyan, and R. Nourmandi-pour, “Comparison of software testing review Black Box and White Box and Gray Box,” *Int. J. Math. Comput. Sci.*, vol. 43, pp. 1442–1447, 2015.

[11] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018, doi: 10.30998/string.v3i2.3048.

[12] T. S. Jaya, “Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung),” *J. Inform. Pengemb. IT*, vol. 3, no. 2, pp. 45–46, 2018, doi: 10.30591/jpit.v3i1.647.