

PENERAPAN ALTERNATIF REKOMENDASI BUKU PADA SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEB

Muhammad Faisal¹, Indah Puji Astuti², Ghulam Asrofi Buntoro³

^{1,2,3} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo

¹ muh.faisal572@gmail.com, ² indahsan.0912@gmail.com, ³ ghulam@umpo.ac.id

Abstrak

Pendataan anggota dan buku di perpustakaan SMA Kyai Ageng Basyaryyah saat ini masih menggunakan sistem manual yang mana pendataan anggota dan koleksi buku masih ditulis dalam buku catatan khusus. Banyaknya buku yang ada di perpustakaan menjadi kendala bagi siswa dalam melakukan pencarian buku. Terlebih lagi ketika seorang siswa mencari suatu buku yang mana belum tentu tersedia apakah masih ada buku tersebut ataukah habis terpinjam, kemudian tidak adanya pengetahuan alternatif buku rekomendasi apabila buku tersebut kosong, sehingga manajemen peminjaman buku tidak terorganisir dengan baik. Maka dari itu diperlukan sebuah sistem untuk menyelesaikan permasalahan tersebut. Penggunaan sistem Informasi perpustakaan berbasis web sangatlah membantu meningkatkan kinerja perpustakaan agar lebih efektif. Dengan menambahkan fitur alternatif rekomendasi buku menggunakan metode similarity pada sistem tersebut dapat mempermudah siswa dan petugas dalam mencari pilihan alternatif apabila buku yang dicari kosong atau tidak ada. Pada penelitian ini sistem dibangun menggunakan SDLC model *prototype*. Pengujian sistem menggunakan beberapa pengujian seperti pengujian kecepatan, fungsionalitas, dan ketepatan dalam menampilkan alternatif buku. Dengan adanya sistem pengelola perpustakaan ini, dapat memudahkan manajemen mulai dari pendataan buku, peminjaman, pencarian buku dan rekomendasi buku sesuai dengan judul yang diinputkan.

Kata kunci : alternatif rekomendasi, perpustakaan, rekomendasi, sistem informasi

1. Pendahuluan

Buku adalah sumber ilmu yang tidak ada habisnya. Terkadang orang haus akan ilmu pengetahuan, sehingga mereka membacanya untuk memuaskan rasa ingin tahunya tentang lingkungan sekitar. Orang bahkan menyebutnya jendela dunia karena semua informasi di dunia dikumpulkan dalam satu buku yang dapat dibaca kapan saja, di mana saja. Namun di Indonesia minat baca masih tergolong rendah. Indonesia berada di peringkat 60 dari 61 negara peserta. Menurut data survei yang dilakukan oleh *Central Connecticut State University* di Amerika Serikat dan diluncurkan pada awal 2017 dengan judul "Negara Paling Sastra di Dunia", menunjukkan rendahnya minat baca orang-orang Indonesia (Tahmidaten and Krismanto 2020).

Salah satu upaya untuk dapat membangkitkan minat baca kepada siswa adalah dengan menyediakan berbagai bahan bacaan variatif, sehingga mendukung pembelajaran dan mendorong kepada siswa untuk menyukai buku. Salah satu media untuk mendapatkan bacaan variatif yaitu, perpustakaan sekolah.

Perpustakaan adalah tempat koleksi buku atau majalah. Walaupun dapat diartikan sebagai koleksi pribadi, namun perpustakaan lebih umum dikenal sebagai sebuah koleksi besar yang dibiayai dan

dioperasikan oleh sebuah kota atau instansi, dan dimanfaatkan oleh masyarakat yang rata-rata tidak mampu membeli buku atas biaya sendiri. Inilah yang disebut dengan perpustakaan konvensional (Astria Firman, Hans F. Wowor 2018). Perpustakaan di SMA Kyai Ageng Basyaryyah masih menggunakan cara manual dalam manajemen seperti pendataan, pencarian dan peminjaman buku.

Perpustakaan berbasis web telah menjadi solusi yang efisien dan inovatif dalam menghadirkan akses mudah dan cepat terhadap berbagai koleksi buku. Namun, tantangan yang dihadapi oleh perpustakaan berbasis web adalah bagaimana menyajikan alternatif rekomendasi buku yang relevan bagi pengguna. Untuk mengatasi tantangan ini, pembuatan sistem informasi perpustakaan dengan menerapkan sistem rekomendasi didalamnya sangatlah menarik dan menjanjikan. Dengan menerapkan sistem rekomendasi pada sistem informasi perpustakaan mempermudah pencarian buku dan dapat menjadi alternatif kedua apabila buku yang dicari kosong atau habis terpinjam.

Sistem informasi (*information systems*) adalah kumpulan elemen terkait yang mengintegrasikan data, memproses informasi, dan membentuk unit untuk menyimpan dan mendistribusikan informasi. Kemudian sistem informasi memiliki komponen

kunci yang terdiri dari *block input*, *block model*, *block output*, *block teknologi*, *block database*, dan *block kontrol*. (Wijaya and Astuti 2019). Menurut (Rahmawati and Bachtiar 2018) sistem informasi terdiri dari lima komponen yaitu *hardware* (perangkat keras), *software* (perangkat lunak), data, prosedur dan manusia.

Sistem Rekomendasi (*recommender system*) adalah teknik yang berfungsi untuk memberi saran kepada pengguna tentang item apa yang sebaiknya dipilih (Rohmah, Himawat Aryadita, and Adam Hendra Brata 2019). Saran ini dapat berkaitan dengan proses pengambilan keputusan untuk berbagai macam aktifitas, seperti barang yang sebaiknya dibeli, musik yang sebaiknya didengar, dan film yang sebaiknya di tonton, dan sebagainya. Rekomendasi sistem diimplementasikan dengan pengembangan media dan aktivitas berbasis online menggunakan aplikasi *web* atau *mobile*. Sistem rekomendasi biasanya merupakan fitur tambahan dari program aplikasi. (Murti and Lestariningsih 2019)

Berdasarkan basis pengetahuan yang digunakan, sistem rekomendasi dibagi menjadi 4 kategori (Alkaff, Khatimi, and Eriadi 2020), yaitu: 1). *Collaborative*: sistem ini hanya melacak *rating* pengguna dan kemudian mengumpulkan pengguna dengan riwayat *rating* yang serupa sehingga rekomendasi dapat dibuat berdasarkan hubungan antara pengguna tersebut. 2). *Content-based*: sistem ini menghasilkan rekomendasi dari fitur yang berkaitan dengan produk tersebut. 3). *Demographic*: sistem ini memberikan rekomendasi produk berdasarkan profil demografi dari pengguna, dan 4). *Mobile based*: Sistem ini menghasilkan rekomendasi produk berdasarkan kesimpulan tentang kebutuhan dan preferensi pengguna. Pengetahuan ini berisi pengetahuan fungsional *eksplisit* tentang bagaimana fitur produk tertentu memenuhi kebutuhan pengguna. Ada beberapa metode yang dapat digunakan dan dipakai untuk membuat rekomendasi alternatif yaitu menggunakan metode *Similarity* menggunakan klausul *SQL LIKE*.

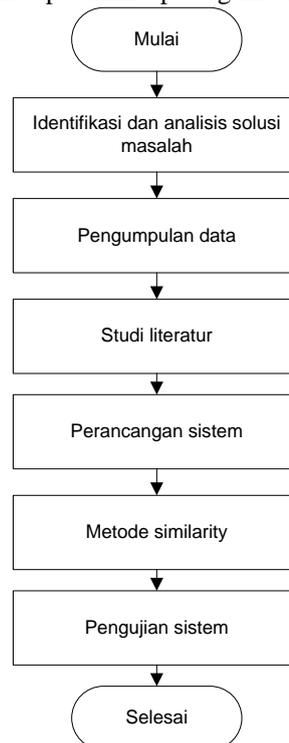
Penelitian sebelumnya yang dilakukan oleh (Alkaff, Khatimi, and Eriadi 2020) yaitu Sistem rekomendasi buku menggunakan *weighted tree similarity* dan *content based filtering* dengan tujuan untuk bagaimana mempermudah pustakawan Provinsi Kalimantan Selatan mencari dan menemukan buku yang berkaitan dengan buku yang sebelumnya. Hasil penelitian berupa sistem informasi yang memiliki fitur rekomendasi buku dengan nilai *precision* sebesar 88%.

Pada penelitian ini akan dibangun sistem Informasi perpustakaan berbasis *web* yang dinamis. *Website* adalah halaman *web* yang saling berhubungan yang biasanya berisi kumpulan informasi dalam bentuk teks, gambar, animasi audio, video, atau kombinasi dari semuanya, yang biasanya dibuat untuk tujuan pribadi, organisasi, dan bisnis. Dari pengertian *website* dapat dibagi menjadi dua

bagian yaitu, statis dan dinamis. Statis ketika konten informasi tetap dan konten informasi disediakan hanya oleh pemilik situs *web*. Situs *web* dinamis konten informasi terus berubah dan dapat diubah oleh pemilik atau pengguna sistem *web* (Jayadi, Mulyawan, and Dolok 2020).

2. Metode Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Tahapan Penelitian

2.1. Identifikasi dan Analisis Solusi Masalah

Ditahap ini, dilakukannya identifikasi dan analisis solusi masalah terhadap permasalahan yang terdapat di perpustakaan SMA Kyai Ageng Basyariyah.

- Identifikasi masalah
Proses manajemen perpustakaan masih dilakukan manual seperti pendataan, pencarian dan peminjaman buku. Petugas dan siswa masih sering kebingungan dalam mencari buku jika ada yang ingin meminjam buku. Cara yang dilakukan selama ini adalah dengan mencari manual di rak buku secara langsung. Hal ini membutuhkan waktu yang cukup lama dan kurang efisien.
- Solusi :
Membuat sistem Informasi perpustakaan yang dapat digunakan untuk peminjaman dan pencarian buku berbasis *web*. Dengan basis *web*, siswa maupun petugas dapat melakukan pencarian, pengelolaan dan peminjaman buku

dengan lebih efisien. Pada pembangunannya akan diterapkan metode *Similarity* untuk alternatif merekomendasikan buku, sehingga petugas atau siswa cukup menginputkan judul buku yang akan dicari. Sistem akan memberikan rekomendasi judul buku sejenis yang mungkin dapat dijadikan rekomendasi buku untuk dipinjam.

2.2. Pengumpulan Data

Pada tahap pengumpulan data, metode pengumpulan data yang digunakan adalah observasi dan wawancara. Observasi dilakukan dengan mengamati langsung kegiatan layanan peminjaman buku di Perpustakaan SMA Kyai Ageng Basyariyah. Wawancara dilakukan melalui tanya jawab langsung dan tatap muka dengan pihak yang terlibat yaitu petugas dan guru perpustakaan SMA Kyai Ageng Basyariyah.

2.3. Studi Literatur

Pada tahap ini, studi literatur dilakukan untuk memperoleh referensi. Referensi diambil dari buku, jurnal, *ebook*, dan halaman *web* terkait permasalahan dan metode yang akan digunakan untuk menyelesaikan masalah yang dihadapi.

2.4. Perancangan Sistem

SDLC (System Development Life Cycle) adalah metode yang digunakan dalam pengembangan perangkat lunak. Pembangunan sistem pada penelitian ini mengadaptasi tahapan-tahapan yang ada pada model pengembangan sistem *prototyping*.

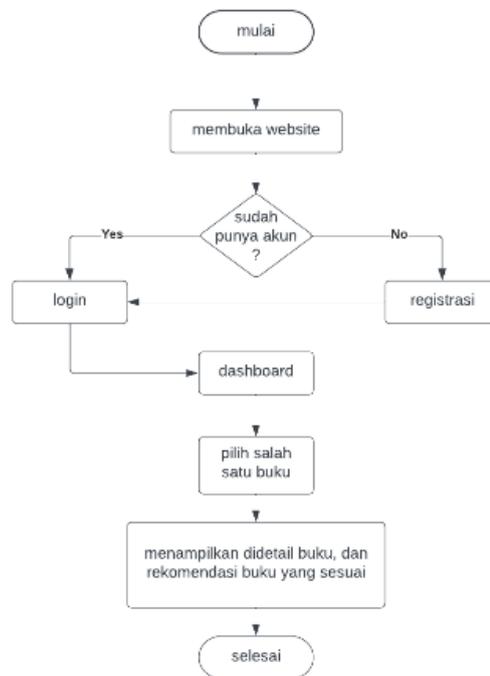
Model *prototyping* merupakan salah satu metode pengembangan sistem dengan cara memberikan contoh penawaran sebuah rancangan kepada orang yang akan menjadi pemakai aplikasi dan memberikan evaluasi *prototype* sebelum dilakukan penulisan sintak (atau tahap *develop*) (Sabaruddin Raja and Fitriani Dewi 2021).

Framework yang akan digunakan untuk sistem Informasi perpustakaan pada penelitian ini adalah *laravel*. *Laravel* adalah sebuah *Framework* untuk membangun *website* berbasis MVC yang ditulis dalam bahasa pemrograman PHP, dirancang untuk meningkatkan kualitas perangkat lunak dan kinerja aplikasi *web* (Kurnia and Aditya 2022). MVC sendiri digunakan untuk memisahkan pengembangan aplikasi berdasarkan komponen utama dari sebuah aplikasi antara lain manipulasi/pengelolaan data, *design* tatap muka, dan pengontrol aplikasi. Oleh karena itu, tidak jarang *Framework Laravel* berbasis *PHP* menjadi favorite dan fondasi dari *Framework* yang memungkinkan para programmer *PHP* untuk membuat aplikasi yang lebih dinamis dan elegan.

Karena *Framework* ini menekankan kesederhanaan dan fleksibilitas dalam desainnya.

Flowchart dan DFD sistem Informasi perpustakaan dapat dilihat pada gambar 2 dan gambar 3.

1. Flowchart Userflows

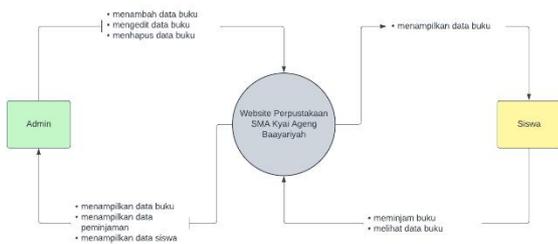


Gambar 2. Flwochart Sistem

Gambar 2 merupakan flowchart dari sistem Informasi perpustakaan, dimulai dari pengguna membuka *website* perpustakaan, terdapat *decision* apakah *user* telah memiliki akun, klik masuk apabila sudah memiliki akun dan klik daftar apabila tidak memiliki akun. *Decision* ini penting karena dalam peminjaman buku nantinya hanya dapat dilakukan oleh *user* yang telah memiliki akun yang emailnya telah terverifikasi. Kemudian langsung diarahkan ke *dashboard*, kemudian klik daftar buku dan pinjam buku yang *user* inginkan. Klik *title* buku apabila *user* ingin melihat detail buku. Pada detail buku tersebut metode *similarity* akan diterapkan. Metode *similarity* ini akan menampilkan buku-buku yang memiliki kemiripan dengan buku yang ada didetail buku tersebut.

2. DFD Level 0

Gambar 3 merupakan *DFD (Data Flow Diagram)* level 0 dalam sistem Informasi perpustakaan. Terdapat 2 *eksternal entity* yaitu admin dan siswa.



Gambar 3. Data Flow Diagram Level 0

3. Struktur Database

Penelitian ini menggunakan *MySQL* sebagai penyimpanan data untuk sistem Informasi perpustakaan SMA Kyai Ageng Basyariyah. *MySQL database* merupakan *software* sistem manajemen basis data *SQL* atau *DBMS* yang *multithread*, dan banyak pengguna yaitu mencapai enam juta instalasi di seluruh penjuru dunia. (Yasin K 2023). Rancangan *database* pada penelitian ini dapat dilihat pada Gambar 4.

| role_has_permissions | |
|----------------------|------------|
| permission_id | bigint(20) |
| role_id | bigint(20) |

| permissions | |
|-------------|--------------|
| id | bigint(20) |
| name | varchar(255) |
| guard_name | varchar(255) |
| created_at | timestamp |
| updated_at | timestamp |

| roles | |
|------------|--------------|
| id | bigint(20) |
| name | varchar(255) |
| guard_name | varchar(255) |
| created_at | timestamp |
| updated_at | timestamp |

Gambar 4. Rancangan Database 1

Gambar 4 adalah sebagian tabel rancangan *database*, terdapat tabel dengan nama *role_has_permission*, *roles*, dan *permission*, kemudian terdapat relasi *many-to-many* pada tabel *permission* dengan *role* yang mana ketika terdapat tabel dengan relasi *many-to-many* harus ada tabel *pivot*, tabel *pivot* disini diberinama dengan *role_has_permission* yang menyimpan id dari masing-masing tabel yang berelasi.

| model_has_roles | |
|-----------------|--------------|
| role_id | bigint(20) |
| model_type | varchar(255) |
| model_id | bigint(20) |

| users | |
|-------------------|--------------|
| id | bigint(20) |
| name | varchar(255) |
| email | varchar(255) |
| email_verified_at | timestamp |
| password | varchar(255) |
| remember_token | varchar(100) |
| created_at | timestamp |
| updated_at | timestamp |

| model_has_permissions | |
|-----------------------|--------------|
| permission_id | bigint(20) |
| model_type | varchar(255) |
| model_id | varchar(255) |

Gambar 5. Rancangan Database 2

Pada gambar 5 dapat dilihat bahwa terdapat tabel user untuk menyimpan data-data dari user, ada id, nama, *email*, kapan *email* diverifikasi, *password* dan lain-lain. Selanjutnya ada tabel *model_has_role* dan *model_has_roles*.

| books | |
|------------------|--------------|
| id | bigint(20) |
| author_id | bigint(20) |
| title | varchar(255) |
| author | varchar(255) |
| publisher | varchar(100) |
| publication date | datetime |
| bookshelf | varchar(255) |
| description | text |

| authors | |
|---------|--------------|
| id | bigint(20) |
| name | varchar(255) |

| borrow_history | |
|----------------|------------|
| id | bigint(20) |
| user_id | bigint(20) |
| book_id | bigint(20) |
| returned_at | datetime |
| admin_id | bigint(20) |
| created_at | timestamp |
| updated_at | timestamp |

Gambar 6. Rancangan Database 3

Pada gambar 6 terdapat tabel inti dari sistem Informasi perpustakaan yaitu tabel *book* yang memiliki *property id*, *author*, *tiile*, *descripsi*, rak, dan lain-lain. Terdapat juga tabel dengan nama *authors* yang memiliki relasi ke tabel *books*, kemudian ada tabel *borrow_history* untuk menyimpan data-data buku yang sedang dipinjam dan buku yang sudah dikembalikan.

2.5. Metode Similarity

Fungsi kemiripan (*Similarity*) adalah langkah-langkah yang digunakan untuk mengidentifikasi kesamaan atau kemiripan antara suatu konten yang dipilih dengan konten-konten yang ada pada *database*. Hampir sama seperti pada metode *cosine Similarity*, Metode *Similarity* atau kemiripan adalah yang digunakan untuk menghitung similaritas dua buah dokumen (Wahyudi 2019).

Similarity adalah langkah yang digunakan untuk mengenali kesamaan atau kemiripan. Dalam penelitian ini menggunakan '*SQL LIKE*' untuk merekomendasikan beberapa buku yang menjadi alternatif untuk dipilih pengguna. Klausa *SQL* digunakan untuk membandingkan nilai dengan nilai yang sama menggunakan operator *wildcard*. Operator *wildcard* sendiri digunakan untuk menggabungkan dengan operator serupa seperti: tanda persen dan garis bawah. Tanda persen digunakan untuk mewakili 0, 1, atau banyak karakter. Garis bawah, di sisi lain, digunakan untuk mewakili satu angka atau huruf. Simbol ini dapat digunakan dalam beberapa kombinasi.

Untuk lebih jelas mengenai metode yang digunakan, dapat dilihat pada gambar 7 dalam bentuk *flowchart*.



Gambar 7. Flowchart Metode Similarity

Berdasarkan gambar 7 dapat dilihat beberapa tahapan yang ada dalam metode *similarity*, yaitu:

- Lakukan pemanggilan data: ditahap ini dilakukan pemanggilan data yang ada didatabase dengan menggunakan *type hint* dengan model *binding*.
- *Split* kalimat: pada bahasa pemrograman *PHP* ada sebuah *method* yang dapat digunakan untuk memecah suatu kalimat menjadi pecahan kata, *method* tersebut bernama *explode*, *methode explode* sendiri membutuhkan 3 parameter (*sparator, string, limiter*).
- Membandingkan data: pada tahap ini dapat dilakukan dengan menggunakan *Klausal SQL like*, membandingkan hasil kata yang berhasil dipecah dengan data yang ada didatabase.
- Menampilkan data yang memiliki kesamaan.

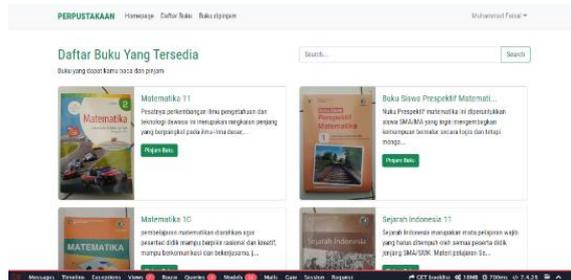
3. Hasil dan Pembahasan

Pada bagian ini akan disampaikan tentang *interface* dan pengujian sistem.

3.1. User Interface Sistem

Berikut adalah beberapa tampilan interface dari sistem Informasi perpustakaan SMA Kyai Ageng Basyariyah.

1. Halaman Daftar Buku



Gambar 8. Halaman Daftar Buku

Pada gambar 8 terdapat halaman daftar buku, di halaman ini terdapat banyak daftar buku dan *form search bar* untuk mencari buku yang *user* inginkan. Untuk cara kerjanya, sistem akan mengambil data yang ada di *url* yang dikirim oleh *user* dengan method '*request*' dan membuat sebuah *decision* yang di dalamnya terdapat *query filtering*, seperti pada gambar 9.

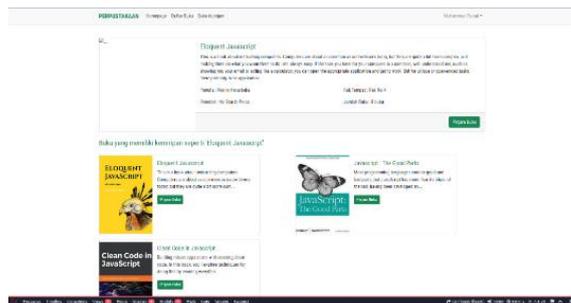
```

// search filter
public function scopeFilter($query)
{
    if(request('search')){
        return $query->where('title', 'like', '%' . request('search') . '%')
            ->orWhere('description', 'like', '%' . request('search') . '%');
    }
}
    
```

Gambar 9. Function Filter Search

Tampilan *booklist* menampilkan daftar buku secara keseluruhan, tetapi apabila *form search bar* terisi halaman daftar buku akan berganti, sesuai yang diinputkan oleh user.

2. Halaman Detail Buku



Gambar 10. Halaman Detail Buku

Pada gambar 10 terdapat halaman detail buku. Pada halaman ini terdapat *cover*, judul, deskripsi lengkap, penulis, penerbit, *quantity*, dan tempat dimana buku diletakkan. Data-data tersebut didapat dari *database* dan sesuai id buku yang pengguna klik. Kemudian dibawah detail *book* terdapat tampilan buku-buku yang memiliki kesamaan *title* dan deskripsi pada detail buku tersebut. Untuk menampilkan sesuai kesamaan pada detail buku menggunakan dan menerapkan metode *Similarity* (kesamaan).

```

public function show(Book $book)
{
    $explode = explode(' ', $book->title);
    $relatedBooks = Book::query();

    foreach ($explode as $word) {
        $relatedBooks->orWhere('title', 'LIKE', '%' . $word . '%');
    }

    $relatedBooks = $relatedBooks->distinct()->get();

    return view('frontend.book.show',[
        'title' => $book->title,
        'book' => $book,
        'relatedBooks' => $relatedBooks,
    ]);
}

```

Gambar 11. Metode Similarity pada PHP

Gambar 11 menjelaskan mengenai metode *Similarity* yang diterapkan, dibuat menggunakan *Framework Laravel* yang telah menerapkan arsitektur *MVC* dengan menggunakan *php v7.4*.

MVC adalah singkatan dari *Model View Controller*, yang merupakan arsitektur untuk membangun sebuah website dengan memisahkan data (*Model*) dari tampilan (*View*) dan memprosesnya (*Controller*). Secara umum, konsep *MVC* adalah arsitektur pemrograman yang memisahkan komponen kunci yang membangun website: manipulasi data, UI pengguna, dan bagian yang mengontrol *website*. (Alip et al. 2022).

Di dalam fungsi *show* membutuhkan sebuah parameter. Parameter tersebut menggunakan *typehint* dengan model *binding Book*. Kemudian dalam *scope* fungsi *show* terdapat *method explode* yang membutuhkan dua parameter, parameter pertama disebut dengan separator dan parameter kedua adalah data yang di *split* (pecah/ bagi). Baris selanjutnya dilakukan iterasi menggunakan *foreach*. *Foreach* digunakan untuk pengulangan sebanyak jumlah kata yang didapat dari pemecahan judul (*title*), didalam *foreach* juga menambahkan fungsi *where title LIKE* untuk kata yang sedang berulang, selanjutnya ketika ada *title* yang sama maka akan diambil dan data tersebut akan di *passing* ke *view*.

3.2. Hasil Pengujian

Pengujian sistem atau *software testing* merupakan kegiatan penting dalam rekayasa perangkat lunak. Pengujian akan menjalankan perangkat lunak dengan tujuan untuk melihat apakah perangkat lunak tersebut sesuai dengan persyaratan yang ditentukan. Sederhananya, aktivitas pengujian akan memproses dan menjalankan program dengan tujuan untuk menemukan *bug*. Pengujian perangkat lunak menjalankan serangkaian proses yang dirancang untuk memverifikasi apa yang seharusnya dan tidak boleh terjadi pada perangkat lunak. (Min, Istiqomah, and Rahmani 2020)

Berikut adalah hasil test yang telah dilakukan pada penelitian ini:

1. Tes Kecepatan dan Tes Ketepatan

Pada test kecepatan (*time request*) dan ketepatan, menggunakan laravel *debug-bar* untuk mengukur kecepatan *time request*, sedangkan untuk mengukur ketepatan menggunakan *dump and die* (*dd()*). *Dump and die* adalah salah satu *function* yang digunakan untuk menguji apakah data yang diproses di dalam *controller* telah berjalan dengan baik. Berikut adalah hasil test yang dilakukan:

- Test Kecepatan

Pada tahap test kecepatan hanya dilakukan pada halaman yang didalamnya menerapkan metode *similarity* yaitu pada halaman *client* bagian detail buku. Tabel hasil pengujian test kecepatan dapat dilihat pada tabel 1.

Tabel 1 Hasil Test Kecepatan

| No | title | Waktu | Kesimpulan |
|----|------------|-------|--------------------|
| 1 | biologi | 412ms | Dari keempat test |
| 2 | matematika | 491ms | yang telah |
| 3 | javascript | 444ms | dilakukan degnan |
| 4 | sejarah | 521ms | masing-masing |
| | | | pencairan kata |
| | | | yang berbeda dapat |
| | | | disimpulkan |
| | | | bahwa waktu yang |
| | | | dibutuhkan untuk |
| | | | memproses |
| | | | metode similarity |
| | | | sangatlah singkat. |
| | | | Pada rata-rata |
| | | | penampilannya |
| | | | menunjukkan |
| | | | angka 467ms. |

- Test Ketepatan

Di tahap ini pengujian dilakukan menggunakan cara *dump and die* pada variabel yang menyimpan hasil proses menggunakan metode *similarity*. Hasil pengujian *dump and die* dapat dilihat pada Gambar 12.

Gambar 12 menunjukkan hasil dari test menggunakan *dump and die* (*dd()*). Pada test tersebut telah dilakukan pengujian pada buku yang berjudul '*Eloquent Javascript*'. Kemudian request yang diminta akan diproses dan menghasilkan beberapa

data dalam bentuk *array* (*array* tersebut berisikan data-data yang memiliki keterkaitan dengan buku berjudul *Eloquent Javascript*). Ada 3 data *array* yang berhasil ditampilkan.

```

Illuminate\Database\Eloquent\Collection {#1407 ▼
  #items: array:3 [▼
    0 => App\Book {#1406 ▼
      #guarded: []
      #connection: "mysql"
      #table: "books"
      #primaryKey: "id"
      #keyType: "int"
      +incrementing: true
      #with: []
      #withCount: []
      #perPage: 15
      +exists: true
      +wasRecentlyCreated: false
      #attributes: array:10 [▼
        "id" => 68
        "author_id" => 30
        "title" => "Eloquent Javascript"
        "description" => ""
        "This is a book about instructing computers. Computers are about as
        common as screwdrivers today, but they are quite a bit more complex,
        and making them do what ▶
        If the task you have for your computer is a common, well-understood one,
        such as showing you your email or acting like a calculator, you can
        open the appropriat ▶
        ""
        "cover" => "assets/covers/V32C56aw9rmSYsa4T7Uhd2r8lh0vgdxiJMoFq68.jpg"
        "qty" => "8"
        "publisher" => "No Starch Press"
        "bookshelf" => "Rak No.4"
        "created_at" => "2022-06-28 07:00:19"
        "updated_at" => "2022-06-28 07:00:19"
      ]
      #original: array:10 [▶]
      #changes: []
      #casts: []
      #dates: []
      #dateFormat: null
      #appends: []
      #dispatchesEvents: []
      #observables: []
      #relations: []
      #touches: []
      +timestamps: true
      #hidden: []
      #visible: []
      #fillable: []
    }
    1 => App\Book {#1405 ▶}
    2 => App\Book {#1404 ▶}
  ]
}
    
```

Gambar 12. Hasil test menggunakan metode *dump and die*

2. Behavioral Testing atau Black Box Testing

Behavioral Testing atau yang dikenal juga sebagai *Black box testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Tes ini dijalankan pada akhir pembuatan perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi dengan baik (Parlika et al. 2020).

Data-data *testing* ini diambil dari *Laravel debugbar*, salah satu fitur yang terdapat pada *Framework Laravel* digunakan untuk mempermudah melakukan monitoring pada aplikasi yang sedang dibangun, seperti tentang *query, route, views, request* time dan lain-lain.

Tabel 2 Hasil Test Behavioral

| No | Kelas Uji | Butir Uji | Hasil Pengujian | Kesimpulan |
|----|--------------|---|-----------------|------------|
| 1 | Login Sistem | Apakah user & admin dapat masuk apabila data sesuai | Sesuai | Normal |

| No | Kelas Uji | Butir Uji | Hasil Pengujian | Kesimpulan |
|----|------------------------|--|-----------------|------------|
| 2 | register | Apakah User & Admin dapat masuk apabila data sesuai | Sesuai | Normal |
| 3 | homepage | Mengecek apakah halaman telah merender dengan baik | Sesuai | Normal |
| 4 | Daftar Buku | Apakah buku-buku yang tersimpan di database telah berhasil ditampilkan | Sesuai | Normal |
| 5 | Detail Buku | Klik pada title buku dan akan diarahkan ke detail buku, apakah telah berjalan. | Sesuai | Normal |
| 6 | Button peminjaman buku | Meng klik button pinjam buku untuk meminjam buku yang diinginkan | Sesuai | Normal |

Dari tabel 2 dapat disimpulkan bahwa pada test *behavioral* ini hanya dilakukan pada *client side* dan tidak untuk *server side*. Program yang telah dibangun tidak ditemukan kendala seperti, *error, loop forever* ketika pemanggilan data, *error n+1* yang sering terjadi di *laravel* dan lain-lain.

4. Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil kesimpulan bahwa dengan adanya *website* untuk sistem informasi perpustakaan di SMA Kyai Ageng Basyariayah dapat mempermudah petugas dalam memanajmen perpustakaan seperti pendataan, pencarian dan peminjaman buku. Selain itu dengan adanya sistem alternatif rekomendasi yang menggunakan metode *similarity* ini, siswa dan

petugas dengan mudah dapat mencari buku alternatif apabila buku yang dicari kosong atau habis terpinjam.

Hasil pengujian menunjukkan bahwa dari segi kecepatan mendapatkan rata-rata waktu 467ms dari 4 sample pengujian, dan untuk pengujian *behavioral* menggunakan *black-box*, semua berjalan dengan baik tanpa ditemui adanya kendala, sedangkan hasil pengujian menggunakan *dump-and-die* dari laravel sistem dapat menampilkan buku yang memiliki keterkaitan dengan judul buku.

Daftar Pustaka:

- Alip, Alip, Sandy Kosasi, I Dewa Ayu Eka Yuliani, Gusti Syarifudin, and David David. 2022. "Implementasi Arsitektur Model View Controller Pada Website Toko Online." *Jurnal Bumigora Information Technology (BITE)* 3 (2): 135–50. <https://doi.org/10.30812/bite.v3i2.1566>.
- Alkaff, Muhammad, Husnul Khatimi, and Andi Eriadi. 2020. "Sistem Rekomendasi Buku Menggunakan Weighted Tree Similarity Dan Content Based Filtering." *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer* 20 (1): 193–202. <https://doi.org/10.30812/matrik.v20i1.617>.
- Astria Firman, Hans F. Wowor, Xaverius Najooan. 2018. "Sistem Informasi Perpustakaan Online Berbasis Web." *Jurnal Teknik Elektro Dan Komputer* 18 (1): 23. <https://doi.org/10.29300/syr.v18i1.1568>.
- Jayadi, Edwin, Bagus Mulyawan, and Manatap Dolok. 2020. "Implementasi Metode Collaborative Filtering Untuk Analisis Data Belanja Konsumen Berbasis Website (Studi Kasus Restoran Mykitchen)." *Jurnal Ilmu Komputer Dan Informasi*, 57–61.
- Kurnia, Yusuf, and Geraldi Aditya. 2022. "Online Learning Service Application Design Using Flutter and Laravel Framework." *Bit-Tech* 4 (3): 109–15. <https://doi.org/10.32877/bt.v4i3.423>.
- Min, Joan Lian, Adila Istiqomah, and Ani Rahmani. 2020. "Evaluasi Penggunaan Manual Dan Automated Software Testing Pada Pelaksanaan End-To-End Testing." *JIT (Jurnal Teknologi Terapan)* 6 (1): 18. <https://doi.org/10.31884/jtt.v6i1.256>.
- Murti, Hari, and Endang Lestariningsih. 2019. "Perancangan Sistem Rekomendasi Buku Pada Katalog Perpustakaan Menggunakan Pendekatan Content-Based Filtering Dan Algoritma FP-Growth." *Proceeding Sintak*, 532–36.
- Parlika, Rizky, Tasya Ardhian Nisaa, Shavira Maya Ningrum, and Berlianda Adha Haque. 2020. "Studi Literatur Kekurangan Dan Kelebihan Pengujian Black Box." *Teknomatika* 10 (02): 131–40.
- Rahmawati, Nurul Alifah, and Arif Cahyo Bachtiar. 2018. "Analisis Dan Perancangan Sistem Informasi Perpustakaan Sekolah Berdasarkan Kebutuhan Sistem." *Berkala Ilmu Perpustakaan Dan Informasi* 14 (1): 76. <https://doi.org/10.22146/bip.28943>.
- Rohmah, Nurur, Himawat Aryadita, and Adam Hendra Brata. 2019. "Pengembangan Sistem Informasi Perpustakaan Berbasis Web Pada Perpustakaan Kecamatan Bungah." *Pengembangan Teknologi Informasi Dan Ilmu Komputer* 3 (3): 2225–34.
- Sabaruddin Raja, and Fitriani Dewi. 2021. "MODEL PROTOTYPESEBAGAI METODE PENGEMBANGAN PERANGKAT LUNAK PADA SISTEM INFORMASI PENGADUAN UMUM (STUDI KASUS : DINAS PERHUBUNGAN PROVINSI KALIMANTAN BARAT)." *KHATULISTIWA INFORMATIKA*, 86–91.
- Tahmidaten, Lilik, and Wawan Krismanto. 2020. "Permasalahan Budaya Membaca Di Indonesia (Studi Pustaka Tentang Problematika & Solusinya)." *Scholaria: Jurnal Pendidikan Dan Kebudayaan* 10 (1): 22–33. <https://doi.org/10.24246/j.js.2020.v10.i1.p22-33>.
- Wahyudi, M Didik Rohmad. 2019. "Penerapan Algoritma Cosine Similarity Pada Text Mining Terjemah Al-Qur'an Berdasarkan Keterkaitan Topik." *Semesta Teknika* 22 (1): 41–50. <https://doi.org/10.18196/st.221235>.
- Wijaya, Yahya Dwi, and Muna Wardah Astuti. 2019. "Sistem Informasi Penjualan Tiket Wisata Berbasis Web Menggunakan Metode Waterfall." *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 273–76.
- Yasin K. 2023. "Pengertian MySQL, Fungsi, Dan Cara Kerjanya (Lengkap)." <https://www.niagahoster.co.id/blog/mysql-adalah/>. May 27, 2023.