

IMPLEMENTATION OF LARAVEL FRAMEWORK IN THE DEVELOPMENT OF LIBRARY INFORMATION SYSTEM (STUDY CASE: SMK PGRI 2 SALATIGA)

Embang Aulia Wicaksono^{1*}; Magdalena A. Ineke Pakereng²

Program Studi Teknik Informatika

Universitas Kristen Satya Wacana

www.uksw.edu

¹672016025@student.uksw.edu,²ineke.pakereng@uksw.edu

(*) Corresponding Author

Abstract—library is a means of support for students to develop their potential by increasing their knowledge through books. good library management is needed to facilitate students to support their learning. SMK PGRI 2 Salatiga has a library. the system applied to manage this library uses conventional methods where all library data is recorded into books. This system has a weakness that is prone to errors in recording and searching data. To deal with this problem, we need a system that can help manage library data. library information system can assist library management because all data can be recorded into the system and accessed through the system. This research was conducted to build a library information system that implements the laravel framework as an application framework. Laravel has various functions that can be used to help web application development. Based on the application testing carried out, the library information system of SMK PGRI 2 Salatiga has been running according to design and can handle library data recording by implementing the functions of the laravel.

Keywords: laravel framework, library information system, eloquent.

Abstrak—Perpustakaan merupakan sebuah sarana penunjang bagi siswa untuk mengembangkan potensi dengan menambah ilmu pengetahuannya melalui buku. diperlukan sebuah pengelolaan perpustakaan yang baik agar dapat memfasilitasi siswa untuk menunjang pembelajaran mereka. SMK PGRI 2 Salatiga memiliki sebuah perpustakaan. sistem yang diterapkan untuk mengelola perpustakaan ini menggunakan metode konvensional dimana seluruh data perpustakaan direkam kedalam buku. sistem ini memiliki kelemahan yaitu rentan terjadi kesalahan dalam perekaman maupun pencarian data. untuk menangani masalah tersebut, maka diperlukan sebuah sistem yang dapat membantu pengelolaan data perpustakaan. sistem informasi perpustakaan dapat membantu pengelolaan

perpustakaan karena seluruh data dapat terekam ke dalam sistem dan dapat diakses melalui sistem. Penelitian ini dilakukan untuk membangun sebuah sistem informasi perpustakaan yang menerapkan framework laravel sebagai kerangka kerja aplikasi. Laravel memiliki berbagai fungsi yang dapat digunakan untuk membantu pengembangan aplikasi web. Berdasarkan dengan pengujian aplikasi yang dilakukan, sistem informasi perpustakaan SMK PGRI 2 Salatiga telah berjalan sesuai dengan rancangan dan dapat menangani perekaman data perpustakaan dengan menerapkan fungsi - fungsi dari laravel.

Kata Kunci: framework laravel, sistem informasi perpustakaan, eloquent.

INTRODUCTION

Library is a facility in educational institutions such as schools which has an important role in the development of science and supporting learning for students (Yulviantoro, 2018). It can be a center for knowledge resources and learning activities that can encourage students to think rationally and find new ideas (Mangnga, 2015). Therefore, good library management is needed to facilitate students to be able to develop themselves.

SMK PGRI 2 Salatiga is a vocational high school in the city of Salatiga which has been established since 1986. This school has a library room managed by the school library staff with one librarian who is in charge of managing and maintaining the library.

At SMK PGRI 2 school library, library staff make various books that are used to record all library activities for one semester. At the end of the semester, all library activities that have been recorded in the book are then included in the library report. At the beginning of each semester, the librarian print books that are used to record library activities. Librarian also need space to store library data books (Aeni et al., 2014).

Manual library management methods can cause errors in data retrieval, such as borrowing and returning books (Nastiti Andharini et al., 2019). Also, there is redundancy in data recording where library data is recorded back into library reports (Wicaksono, 2020). Therefore, web-based information systems can be a solution to assist the librarian in managing library data (Puspitasari, 2016). With the library information system, the entire library activity process will be recorded in the system (Hutagalung & Arif, 2018). So the librarian can more easily manage library data.

There are several previous studies used as a reference in designing the library information system of SMK PGRI 2 Salatiga. A study entitled "Designing of Library Information System to Support Learning in High School" (Wardhana, 2018) designed a system that helps improve performance in managing libraries, starting from borrowing data management, book data, to making school library reports by providing a dashboard for the librarian.

Then the study entitled "Development of a Robust Library Management System" (Iwayemi & Oyeniyi, 2019) designed a library management system for the Federal Polytechnic of Ile-Oluji, Nigeria. This study designed a web application based on the PHP programming language. The system assists in the circulation of book lending, book collection management, and has 3 actors who can use the system, namely students, staff, and admin (Iwayemi & Oyeniyi, 2019).

Another study entitled "Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web pada Smk Citra Negara Depok" (Hutagalung & Arif, 2018) designed a library information system that has an additional feature to recording library visitor data into the system by entering data into the provided page and making library reports. Loan transactions in the system are accommodated in the loan table that is connected to the book and member table by intermediate tables (Hutagalung & Arif, 2018).

Based on those studies, this study designed a library information system to meet the needs of SMK PGRI 2 Salatiga library. However, in this study, the library information system was built with the MVC (Model View and Controller) architecture.

MVC is an application architecture that consists of three layers including the Model layer which interacts with the database, the View layer for the application interface, and the Controller layer which manages the other two layers (Majeed & Rauf, 2018). The MVC architecture has the advantage that the division of applications by class allows the use of reusable code (Uyun & Rifqi, 2010).

To form an MVC architecture in an application, this study applies an MVC architecture

web application development framework named Laravel (Verma, 2014). This framework was chosen because Laravel uses a small number of configuration files compared to other frameworks, making it easier to start an application development (Armel, 2014). This research was conducted to build a library information system for SMK PGRI 2 Salatiga by implementing Laravel as an application development framework.

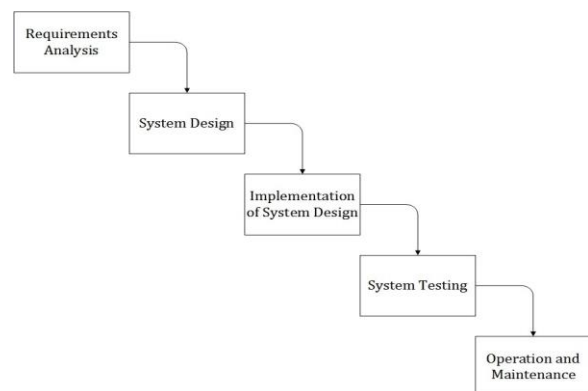
MATERIAL AND METHOD

Data Collection Method

In this study, the data needed to create a library information system was collected using the observation method to case studies to determine field conditions. Then find out the problems that must be answered in the study by interview the library staff. Furthermore, conducting literature studies to find the information needed to create a library information system.

System Development Method

In the development of this library information system, the SDLC (System Development Life Cycle) method is used with the Waterfall model. The Waterfall model is an application development model with stages that are carried out sequentially as in Figure 1. This model is suitable for projects of this scale, and any mistakes and shortcomings can be corrected before moving on to the next stage (Kannan et al., 2014). This model has 5 stages in the application development process, namely, the analysis stage, the design stage, the implementation stage, the testing phase, and the operational and maintenance stage (Kannan et al., 2014).



Source: (Wicaksono, 2020)

Figure 1 Waterfall Model Stages

The stages of designing a library information system using the waterfall model are as follows: Requirement analysis stage. This stage is the stage of analyzing the needs of the system to be designed by conducting interviews with the school librarian.

Design Stage. this stage is the system design stage which will refer to the needs analysis that has been done previously. Implementation stage. At this stage, the results of the previously created designs are then implemented into the code in the Laravel framework. The system database structure design is created using a Laravel function called Migrations(Armel, 2014). This function is used by creating a database migration file created using artisan, then migrating it to MySQL with artisan (Armel, 2014). Artisan is Laravel's command-line interface which allows you to perform functions such as creating controllers, models, migrations, generating dummy data, and many others. (Armel, 2014).

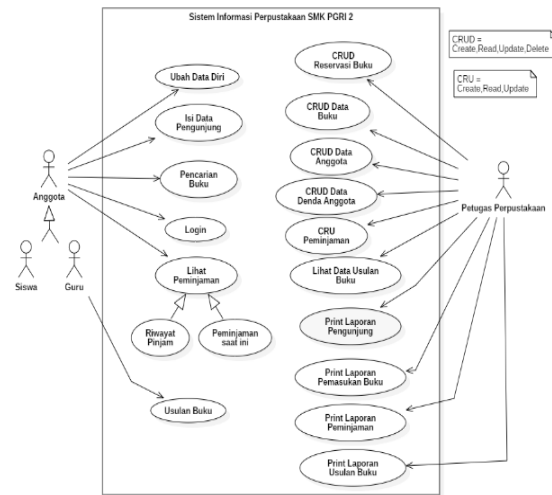
System testing stage. At this stage, the system that has been created is tested to find bugs and errors using the black box method. This method is a method of testing system functionality based on specifications and focuses on the input and output generated from the system (Nidhra & Dondeti, 2012).

Operation and maintenance stage. This stage is the last stage of the library information system development. At this stage, the system is ready to be installed and implemented in the library of SMK PGRI 2 Salatiga.

5. The library clerk manages the library visitor report data
6. Every library member who enters the library must fill in the library visitor data
7. teachers can propose additional collections of library books for teaching and learning purposes
8. The librarian manages additional collections of library books proposal.
9. The librarian notes the library members fines
10. Library members can search for a book in the library
11. Library members can see their loan status

B. Designing The System

1. Information System Design Diagrams



Source: (Wicaksono, 2020)
Figure 2 Use Case Diagram of Library Information System

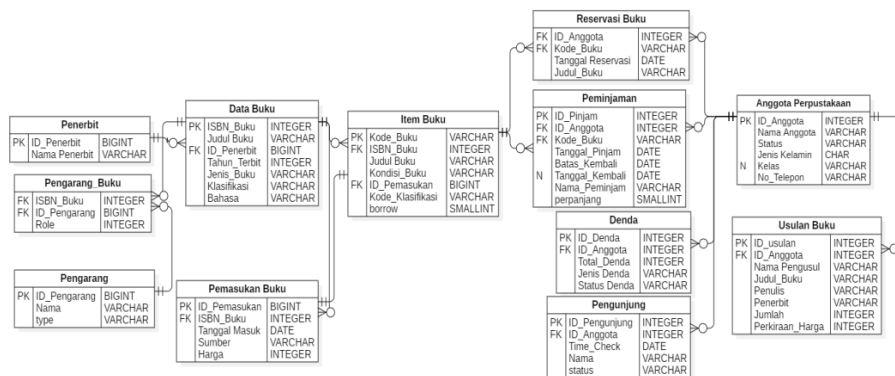
Figure 2 shows that the library information system of SMK PGRI 2 Salatiga has 2 main actors who can run the system, namely member and librarian.

RESULT AND DISCUSSION

A. Requirement Analysis

Based on the results of interviews and observation at the school library, the following requirement analysis results are obtained.

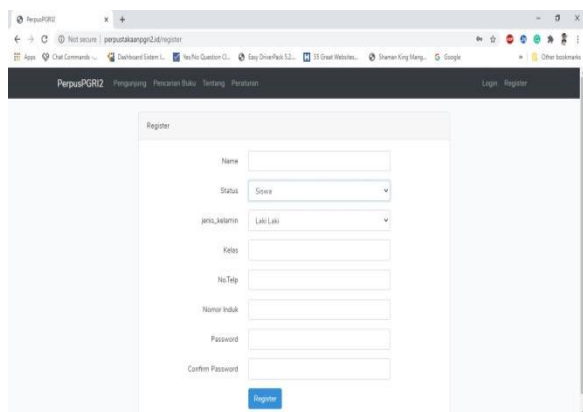
1. The librarian can manage book lending data of library members.
2. The librarian recapitulates book loan data for library reports.
3. The librarian can manage library book collection data.
4. The librarian manages the book-entry report data



Source: (Wicaksono, 2020)
Figure 3 Entity Relational Diagram of Library Information System


```
$this->middleware('guest')->except('logout');
}
public function username() {
return 'username';
}
}
```

The controller login function was added so that laravel can log in using the username which in this case is the registration number. In the redirectTo function, logged in users will be checked for their status attribute. If the status of the user is admin, it will be redirected to the admin dashboard page. If not, then the user will be redirected to the main page for library members.



Source: (Wicaksono, 2020)
Figure 6 Register User Page

Figure 6 shows the library member account registration page. Data sent via registration form will be validated in the validator function. If an attribute is found that does not meet the conditions, the process will be redirected back to the registration page and then displays a warning to the user. Data that has successfully passed validation will be saved to the database using Eloquent ORM by creating a member or user table object in the create function. Eloquent ORM (Object Relational Mapper) is Laravel's active record implementation which is used to interact with the database and perform CRUD (Create, Read, Update, Delete) without using SQL(Parkar et al., 2016). The following is a controller class for validating and storing registration data.

```
<?php
namespace App\Http\Controllers\Auth;
use App\User;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
use RegistersUsers;
protected $redirectTo = '/home';
public function construct() {
```

```
$this->middleware('guest');
}
protected function validator(array $data) {
return Validator::make($data, [
'name' => ['required', 'string', 'max:255'],
'status' => ['required', 'string'],
'jenis_kelamin' => ['required'],
'kelas' => ['string', 'nullable'],
'telepon' => ['numeric', 'digits_between:10,13'],
'username' => ['required', 'string', 'max:255', 'unique:users'],
'password' => ['required', 'string', 'min:8', 'confirmed'],
]);
}
protected function create(array $data) {
return User::create([
'name' => $data['name'],
'status' => $data['status'],
'jenis_kelamin' => $data['jenis_kelamin'],
'kelas' => $data['kelas'],
'telepon' => $data['telepon'],
'username' => $data['username'],
'password' => Hash::make($data['password']),
]);
}
```

Eloquent uses model classes as objects to hold data. Therefore, for the model class to be consistent with the existing tables in the database, it is given a function that states the same constraints as the database table. the program code for the publisher table model is as follows:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class publisher extends Model{
public function book(){
return $this->hasMany('App\Book', 'ID Penerbit');
}
}
```

In the Entity Relational Diagram of the Library Information System, the book table only depends on the publisher table by retrieving the foreign key id from the publisher table. Therefore, a book function was created which states that the publisher model has multiple book models using the HasMany function. The first parameter is the directory location of the related model, the second parameter is the foreign key of the related model. While the program code of the book model is as follows:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Book extends Model {
protected $primaryKey = 'ISBN';
public $incrementing = false;
protected $keyType = 'string';
public function publisher() {
return $this->belongsTo('App\publisher', 'ID Penerbit');
}
```

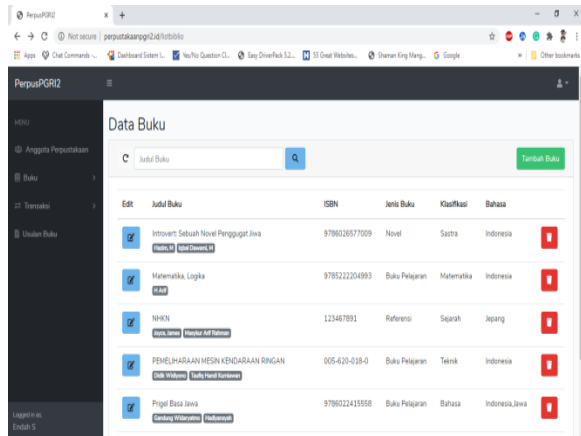
```

public function book_item() {
return $this-
>hasMany('App\book_item', 'ISBN', 'ISBN');
}
public function book_entry() {
return $this-
>hasMany('App\book_entry', 'ISBN', 'ISBN');
}

public function author() {
return $this-
>belongsToMany('App\author', 'book_authors', 'book_id', 'author_id')
->withPivot('role');
->withTimestamps();
}
}

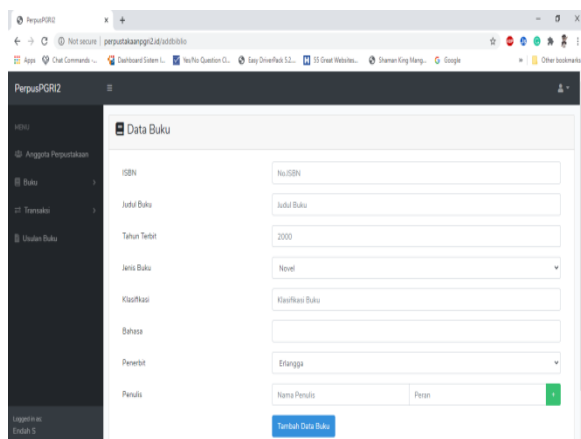
```

book model has publisher function which states that the book model depends on the publisher model using belongsTo function. The first parameter is the location of the related model, the second parameter is the foreign key of the publisher table in the book table. Laravel will automatically detect the primary key used by the parent table with the name *id*. However, if the names are different, then the HasMany and belongsTo functions can be added to the third parameter to determine the primary key used.



Source: (Wicaksono, 2020)

Figure 7 Book Data List Page



Source: (Wicaksono, 2020)

Figure 8 Add Book Data Page

Figure 7 shows the book data list page accessed from the admin dashboard. On this page, the librarian can delete, add, modify and search book data. To add book data, the librarian can press the “tambah buku” button, then it will be redirected to the add book form page as shown in Figure 8. Data from the add book form is sent to the Admin book controller in the insertbiblio function to add book data. The following is the program code for the insertbiblio function on the admin book controller:

```

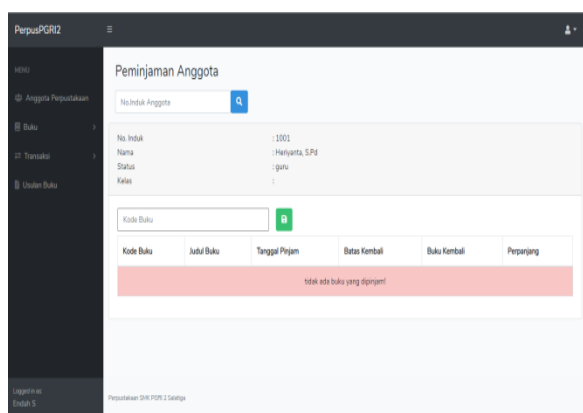
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Collection;
use Illuminate\Support\Facades\Input;
use Maatwebsite\Excel\Facades\Excel;
use App\Exports\BookEntry;
use App\Exports\Visitorlist;
use App\Exports\LoanReport;
use App\publisher;
use App\author;
use App\Book;
use App\book_entry;
use App\book_item;

class AdminbookController extends Controller
{
public function insertbiblio(Request $request
) {
$this->validate($request, [
'ISBN' => 'required|numeric', 'Judul' => 'required',
'Tahun_Terbit' => 'required|numeric|digits:4'
,
'Klasifikasi' => 'required',
'Bahasa' => 'required' ]]);
$Book = new Book;
$Book->ISBN = $request->input('ISBN');
$Book->{'Judul Buku'} = $request-
>input('Judul');
$Book->{'Tahun Terbit'} = $request-
>input('Tahun_Terbit');
$Book->{'Jenis Buku'} = $request-
>input('Jenis Buku');
$Book->Klasifikasi = $request-
>input('Klasifikasi');
$Book->Bahasa = $request->input('Bahasa');
$Book->{'ID Penerbit'} = $request-
>input('Penerbit');
$Book->save();
for ($i=0; $i < count($request-
>input('Penulis')); $i++) {
$author_id = author::firstOrCreate(['nama' =>
$request-
>input('Penulis.' . $i)], ['type' => 'Nama Pribadi']);
$Book->author()-
>attach($author_id, ['role' => $request-
>input('role.' . $i)]);
}
return redirect('/addbiblio')->with
('success', 'Data Berhasil Dimasukkan');
}
}

```

The insertbiblio function has the following program flow. First, all attribute request data will be validated one by one. Then the Book model is initialized to an object variable named \$book. Each attribute in \$book is assigned a value based on the incoming request from the add book form. after that, the data from \$book is stored using syntax save().

To save author data from the book, it is done in a for loop as many as the number of author input obtained. First, the name of the author that is entered is searched in the database and entered into the \$author_id object variable. If the data you are looking for is not found, a new record will be created. Then, the \$author_id and \$book data are linked and stored in the database.



Source: (Wicaksono, 2020)

Figure 9 Book Loaning Page

Figure 9 shows the interface of the book loaning page on the admin dashboard. On this page, the librarian can make borrowing transactions by entering the member's registration number, then the librarian enters the book code that will be borrowed by the library members. Books that have been borrowed will be displayed in the table below the member information. The process of borrowing this book is done on the TransactionController. The functions used for the TransactionController are as follows:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\User;
use App\fine;
use App\book_item;
use App\loan;
use App\reservation;
use Carbon\Carbon;
class TransactionController extends Controller {
public function __construct() {
$this->middleware('auth.admin');
}
public function transactionloan() {
return view('admin.loan');
}
```

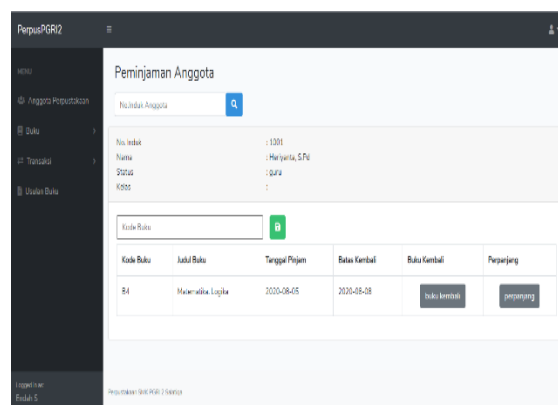
```
}

public function showmemberloan(Request $request) {
$user = User::where('username',$request->id)->first();

if (!isset($user)) {
return redirect()->back()->with('error','nomor induk tidak ditemukan');
}
return view('admin.loan')->with('user', $user);
}

public function addloan(Request $request) {
$book_item = book_item::findOrFail($request->input('kode'));
if ($book_item->borrow == 1) {
$user = User::findOrFail($request->input('id'));
$today = Carbon::now()->format('Y-m-d');
$returnday = Carbon::now()->addDays(3)->format('Y-m-d');
$book_item->borrow = 0;
$book_item->save();
$user->book_item()->attach($book_item, ['nama peminjam' => $user->name,'kode buku' => $book_item->{'kode buku'},'judul buku' => $book_item->{'judul buku'},'tanggal pinjam' => $today, 'batas kembali' => $returnday, 'perpanjang' => 0]);
} else {
return redirect()->back()->with('error','buku sedang dipinjam');
}
return redirect()->back();
}
```

The showmemberloan function is used to search for library member data based on the entered registration number. The addloan function is used to enter loan data from library members into the database. The book codes are searched in the database and then see whether the status can be borrowed. If possible, the book loan data is added to the database and then the status of the book is changed so that it cannot be borrowed by other people. A successful transaction can be seen in Figure 10.



Source: (Wicaksono, 2020)

Gambar 10 Library Member Loan Data

D. System Testing

At the system testing stage, SMK PGRI 2 Salatiga Library Information System conducted a system functionality test using the black box method. The following are the results of black-box testing on the process of adding book data and borrowing books which can be seen in Table 1 and Table 2.

Table 1. Add Book Data Function Testing

Scenario Condition	Expected Results	Received Result	Conclusion
The book data form is filled following the provisions	Data saved in database	Data successfully saved to database	success
Some attributes are not filled in the book data form	A warning appears on the text field that is not filled	"data dibutuhkan" warning appears in the text field	success
The ISBN attribute is filled in with letters	A warning appears on the ISBN text field	A warning "ISBN harus angka" appears in the ISBN text field	success

Source: (Wicaksono, 2020)

Table 2. Book Loan Function Testing

Scenario Condition	Expected Results	Received Result	Conclusion
The registration number is filled following existing data	Display library member data and borrowed books	Display library member data and borrowed books	success
Registration number filled randomly	data not found warning appears	A "the identification number not found" warning appears	success
Book code is filled in according to existing data	Data has been successfully stored and displayed in the table	Data has been successfully stored and displayed in the table	success
Book code filled randomly	A data not found warning appears	A warning "the identification number not found"	success

Source: (Wicaksono, 2020)

Based on Table 1, the book loan function manages to save form data into the database. The validation function for the attribute can give a warning to the user if it does not meet the existing conditions.

In Table 2, the member registration number search function can find the data registered in the database. If the registration number is not registered in the database, a warning will appear that the data was not found. If the book codes are not registered in the database, a warning will appear that the data was not found.

E. Operation and Maintenance

The information system application made is offered to SMK PGRI 2 Salatiga to be implemented in the school library. The following are the specifications used:

1. Hardware System

System Manufacturer: To Be Filled By O.E.M.
 System Model: To Be Filled By O.E.M.
 BIOS: (04/24/2014)ZX-G41D3LM Ver:2.1
 Processor: Intel(R) Core(TM)2 Duo CPU E6550
 Memory: 2048MB RAM
 Available OS Memory: 2014MB RAM

2. Display System

Card name: Intel(R) Q45/Q43 Express Chipset
 Chip type: Intel(R) 4 Series Express Chipset Family
 Display Memory: 782 MB
 Dedicated Memory: 32 MB
 Shared Memory: 750 MB
 Current Mode: 1024 x 768 (32 bit) (60Hz)
 Monitor Name: Generic PnP Monitor

3. Software

Operating System: Windows 7 Ultimate
 Web Server: Apache
 Database: MySQL
 Framework: Laravel 5.8.38
 UI Framework: Bootstrap 4.1.0
 Web Browser: Google Chrome

4. Web Hosting

Storage: 2GB
 Bandwidth: Unlimited
 Database: MySQL Support

CONCLUSION

This research has made a library information system that implements the Laravel web development framework and has been running

according to the design. The book loan function can handle the data of each library member. Eloquent ORM provides convenience in book loan transactions with a code structure that is easy to read and build. Also, model class objects that are interconnected with predefined constraint functions based on database structure prevents conflict in data storage. Laravel makes it easy to create user authentication functions that can be modified as needed. The database structure can be created with the migration function in the Laravel framework without having to interact directly with the database.

REFERENCE

- Aeni, W. N., Santosa, S., & Supriyanto, C. (2014). Algoritma Klasifikasi data mining naïve bayes berbasis Particle Swarm Optimization untuk deteksi penyakit jantung. *Jurnal Pseudocode*, 1(1), 11–14. <https://ejournal.unib.ac.id/index.php/pseudocode/article/view/57/>
- Armel, J. (2014). Web application development with Laravel PHP Framework version 4 [Web application development with Laravel PHP Framework version 4]. In *Media engineering* (Issue April). <https://www.theseus.fi/handle/10024/74052>
- Hutagalung, D. D., & Arif, F. (2018). RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEB PADA SMK CITRA NEGARA DEPOK. *Jurnal Rekayasa Informasi*, 7(1), 13–22. <https://doi.org/10.1017/CBO9781107415324.004>
- Iwayemi, A., & Oyeniyi, S. (2019). Development of a Robust Library Management System. *International Journal of Computer Applications*, 178(12), 9–16. <https://doi.org/10.5120/ijca2019918850>
- Kannan, V., Jhahharia, S., & Verma, S. (2014). Agile vs waterfall: A Comparative Analysis. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 3(10), 2680–2686. <http://ijsetr.org/wp-content/uploads/2014/10/IJSETR-VOL-3-ISSUE-10-2680-2686.pdf>
- Majeed, A., & Rauf, I. (2018). MVC Architecture: A Detailed Insight to the Modern Web Applications Development. *Crimson Publishers*, 1(1), 1–7. <https://crimsonpublishers.com/prsp/pdf/PRSP.000505.pdf>
- Mangnga, A. (2015). Peran Perpustakaan Sekolah Terhadap Proses Belajar Mengajar Di Sekolah. *Jupiter*, 14(1), 38–42. <https://journal.unhas.ac.id/index.php/jupiter/article/view/27/>
- Nastiti Andharini, S., Puji Lestari, N., Ratna Satiti, N., & Roz, K. (2019). Analysis of Library Management Information System in Muhammadiyah 2 Vocational High School, Malang. *KnE Social Sciences*, 3(13), 948. <https://doi.org/10.18502/kss.v3i13.4259>
- Nidhra, S., & Dondeti, J. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, 2(2), 29–50. <https://doi.org/10.5121/ijesa.2012.2204>
- Parkar, V. V, Shinde, P. P., Gadade, S. C., & Shinde, P. M. (2016). Utilization of Laravel Framework for Development of Web Based Recruitment Tool. *National Conference On "Changing Technology and Rural Development,"* 36–41. <https://www.iosrjournals.org/iosr-jce/papers/Conf.16051/Volume-1/8.36-41.pdf?id=7557>
- Puspitasari, D. (2016). Sistem informasi perpustakaan sekolah berbasis web. *Pilar Nusa Mandiri*, 12(2), 227–240. <http://ejournal.nusamandiri.ac.id/index.php/pilar/article/view/277>
- Uyun, S., & Rifqi, M. (2010). Implementation of Model View Controller (MVC) Architecture on Building Web-Based Information System. *Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*, 47–50. <https://journal.uui.ac.id/Snati/article/view/1939/>
- Verma, A. (2014). MVC Architecture : a Comparative Study Between Ruby on Rails and Laravel. *Indian Journal of Computer Science and Engineering (IJCSE)*, 5(5), 196–198. <http://www.ijcse.com/docs/INDJCSE14-05-05-053.pdf>
- Wardhana, A. (2018). Designing of Library Information System to Support Learning in High School. *International Journal of Computer Techniques*, 5(1), 1–4. <http://www.ijctjournal.org/Volume5/Issue1/IJCT-V5I1P1.pdf>

Wicaksono, E. A. (2020). *Laporan Akhir Penelitian - Implementasi Framework Laravel dalam Pembangunan Sistem Informasi Perpustakaan SMK (Studi Kasus: SMK PGRI 2 Salatiga)*.

Yulviantoro, A. B. (2018). Pengembangan Sistem Informasi Perpustakaan Sekolah Berbasis Web Di Smk Negeri 1 Jogonalan [Universitas Negeri Yogyakarta]. In *Universitas Negeri Yogyakarta*.
[https://eprints.uny.ac.id/59325/1/naskah skripsi fix.pdf](https://eprints.uny.ac.id/59325/1/naskah_skripsi_fix.pdf)