

A. Create

```
public function store(StoreBookRequest $request)
{
    $validatedData = $request->validate([
        'title'=>'required',
        'synopsis'=>'required',
        'image' =>'image|file|max:1024'
    ]);
    if($request->file('image')){
        $validatedData['image']= $request->file('image')->store('book_images');
    };
    Book::create($validatedData);
    return redirect('/')->with('sucess','Buku Berhasil Ditambahkan');
}
```

Potongan kode diatas merupakan potongan kode yang berfungsi untuk menyimpan data buku baru. Fungsi ini bekerja dengan melakukan validasi terhadap data yang dikirimkan dari halaman, kemudian memeriksa apakah ada *file* gambar yang diupload. Setelah semua data divalidasi maka data akan dimasukkan ke dalam table *books*.

```
57 public function store(StoreReservationRequest $request)
58 {
59     $reservation = new Reservation();
60     $reservation->id_pinjam = $request->id_pinjam;
61     $reservation->user_id = $request->borrower_id;
62     $reservation->total_books = count($request->book_id);
63     $reservation->borrow_date = Carbon::now();
64     $reservation->must_return_date = Carbon::now()->addDay(7);
65     $reservation->save();
66     foreach($request->book_id as $book){
67         $details = new reservation_detail();
68         $details->reservation_id = $reservation->id;
69         $details->book_id = $book;
70         $details->save();
71
72         $book= Book::findorfail($book);
73         $update = [$book->is_borrowed = true];
74         $book->update($update);
75     };
76     return redirect('/pinjambuku')->with('success','Data berhasil ditambahkan');
77 }
78
```

Potongan kode diatas merupakan potongan kode yang berfungsi untuk menyimpan data baru peminjaman buku. Baris ke 59 hingga baris ke 65 digunakan untuk menambahkan data ke tabel *reservation* kemudian pada baris ke 66 hingga 75 dilakukan perulangan untuk menambahkan buku yang dipinjam ke tabel *reservation_details* dan juga melakukan perubahan pada kolom *is_borrowed* yang awalnya bernilai *false* menjadi *true* pada tabel *books* agar buku tidak bisa dipinjam selama belum dikembalikan.

B. Update

```
90 public function update(UpdateBookRequest $request)
91 {
92     $validatedData = $request->validate([
93         'title'=>'required',
94         'synopsis'=>'required',
95         'image' =>'image|file|max:1024'
96     ]);
97     if($request->file('image')){
98         if($request->oldImage){
99             Storage::delete($request->oldImage);
100         }
101         $validatedData['image']= $request->file('image')->store('book_images');
102     };
103     Book::where('id',$request->id)
104     ->update($validatedData);
105     return redirect('/')->with('success','Data Berhasil Diubah');
106 }
```

Potongan kode diatas digunakan untuk melakukan perubahan data buku.

```
public function update($id)
{
    $reservation= Reservation::findorfail($id);
    $reservation->return_date = Carbon::now();
    $reservation->save();

    $details = reservation_detail::all()->where('reservation_id',$id);
    foreach($details as $detail){
        $book = Book::findorfail($detail->book_id);
        $book->is_borrowed = false;
        $book->save();
    }
    return redirect('/listpinjambuku');
}
```

Potongan kode diatas digunakan untuk melakukan perubahan pada status peminjaman buku. Kode ini merubah tabel *reservation* kemudian juga tabel *books*.

C. Delete

```
107 public function delete($id){
108     $book= Book::find($id);
109     $book->delete();
110     return redirect('/')->with('success','Data Berhasil Dihapus');
111 }
```

Potongan kode ini digunakan untuk menghapus data buku tetapi hanya secara logical saja.