

Tugas Praktikum VII

(Kelompok 8)

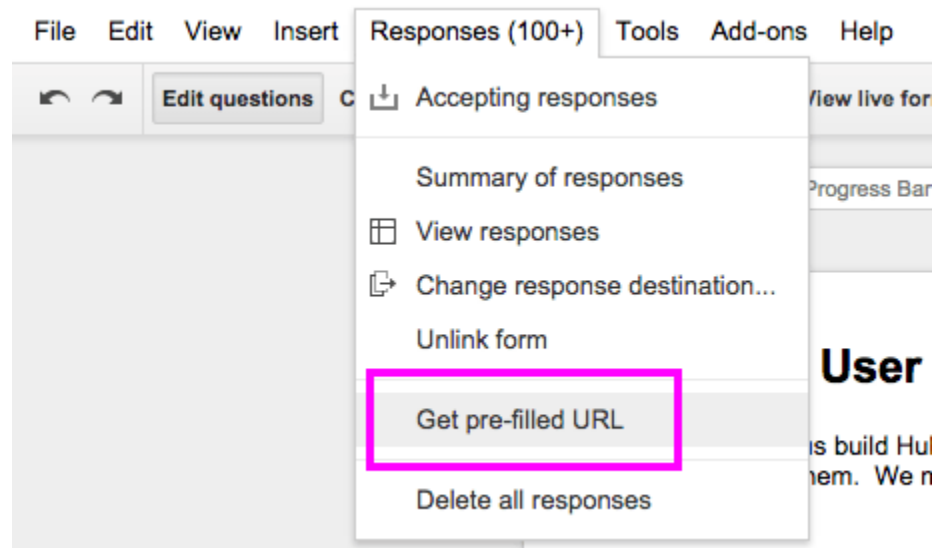
- M. Naufal Muafa (1301180091)
- Ingrid Resmi Benita (1301184107)
- Saskia Putri Ananda (1301184157)
- Anis Novitasari (1301184424)

1. Banyak aplikasi web saat ini yang memanfaatkan API dari web lain. Misalnya:

- ketika mengisi form registrasi, memanfaatkan API dari Google untuk mengisi form otomatis dengan data-data pengguna yang ada di Google
- mengisi kelengkapan data alamat pengguna, dengan data IP Address (atau lokasi GPS android)

a. Cari info bagaimana mengakses API Google tersebut

- Buka halaman edit untuk Google Form, kemudian klik response. Pilih "Get pre-filled URL"



- Isi form dengan merge tag yang diinginkan

Answer questions you want to pre-fill, then click submit.

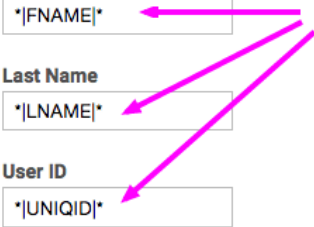
User Validation

First Name
|FNAME|

Last Name
|LNAME|

User ID
|UNIQID|

Enter Mailchimp Merge Tag names into answer pre-fill boxes.



- Salin Pre-filled google form URL

Share the link to this form with your pre-filled responses.

<https://docs.google.com/forms/d/1MnRQJ3XZxsRBSVp28M7SmBLooHZONoG0n1M8Gm5jniE/viewf>

Press ⌘+C to copy.

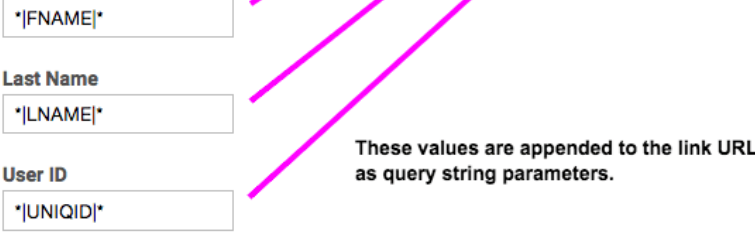
User Validation

First Name
|FNAME|

Last Name
|LNAME|

User ID
|UNIQID|

These values are appended to the link URL as query string parameters.



- Implementasikan dalam sistem

b. Bagaimana memanfaatkan data IP Address (atau GPS Android) untuk melengkapi data alamat

Untuk mendapatkan data lokasi pengguna, developer dapat menggunakan library `com.google.android.gms.location`
Kemudian, data lokasi pengguna bisa ditampilkan. Kode berikut menampilkan data lokasi pengguna pada logcat

```
client.getLastLocation().addOnSuccessListener(MainActivity.this,
    new OnSuccessListener() {
        @Override
        public void onSuccess(Location location) {
            if (location != null) {
                Log.d("TAG", location.toString());
            }
        }
    });
```

2. a. Pilih bahasa pemrograman OO untuk implementasi program TUBES. Jelaskan alasan pemilihan Anda.

Kami memilih bahasa pemrograman PHP untuk implementasi tubes. Kami memilih bahasa PHP karena cocok digunakan untuk membangun sistem kami. Kami juga pernah mempelajari bahasa PHP di mata kuliah pemrograman web.

b. Download compiler bahasa pemrograman tsb, dan install.

```
Microsoft Windows [Version 10.0.18363.1316]
(c) 2019 Microsoft Corporation. All rights reserved.

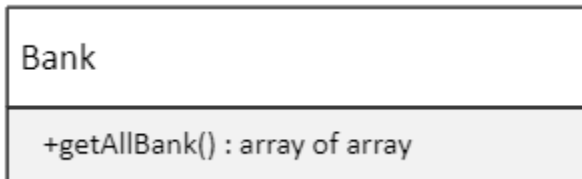
C:\Users\naufa>php -v
PHP 7.3.11 (cli) (built: Oct 22 2019 11:20:18) ( ZTS MSVC15 (Visual C++ 2017) x64 )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.11, Copyright (c) 1998-2018 Zend Technologies

C:\Users\naufa>
```

3. Implementasikan setidaknya tiga class dari Class Diagram yang ada di DPPL Anda dengan bahasa pemrograman

yang telah Anda install :

a. Class paling sederhana (paling sedikit atribut & methodnya)



```
Bank.php X
app > Bank.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Support\Facades\DB;
6
7  class Bank
8  {
9      public static function getAllBank()
10     {
11         return DB::table('banks')->select('id', 'name')->get();
12     }
13 }
```

b. Class yang sedang (jumlah atribut & methodnya "rata-rata")

SampleEbookFile

#table : string

#primaryKey : string

+getNewSampleEbookFilesId() : int

+getSampleEbookFile(sampleEbookFileId) : array

+getSampleBookFileURL(bookId) : string

SampleEbookFile.php X

app > SampleEbookFile.php

```
1  <?php
2  |
3  namespace App;
4
5  use Illuminate\Support\Facades\DB;
6  use App\Book;
7  use Illuminate\Database\Eloquent\Model;
8
9  class SampleEbookFile extends Model
10 {
11     protected $table = 'sample_ebook_files';
12
13     protected $primaryKey = 'id';
14
15     public static function getNewSampleEbookFilesId()
16     {
17         return SampleEbookFile::get()->count() + 1;
18     }
19
20     public static function getSampleEbookFile($sampleEbookFileId)
21     {
22         return DB::table('sample_ebook_files')
23             ->select('id', 'name')
24             ->where('id' , $sampleEbookFileId)
25             ->first();
26     }
27 }
```

```
28     public static function getSampleBookFileURL($bookId)
29     {
30         $sampleEbookFileId = Book::getSampleEbookFileId($bookId);
31         $sampleEbookFile = SampleEbookFile::getSampleEbookFile($sampleEbookFileId);
32         return "/ebook/sample_ebook_files/".$sampleEbookFile->id."/". $sampleEbookFile->name;
33     }
34 }
35
```

- c. Class yang paling kompleks (paling banyak atribut & methodnya, atau mengandung kode untuk pengksesan API soal nomor 1, atau yang mengandung aspek "kecerdasan buatan" (jika ada))

Book

table : string

fillable : array of string

+store() : void

+getNewBookId() : int

+getBookDataForDashboardPublisher() : array

-convertPriceToCurrencyFormat(price) : String

+getBook(bookId) : array

+updateBook(book) : void

+updateCoverPhoto(photoId,bookId) : void

+uploadSampleEbook(file,bookId) : void

+deleteBook(id) : void

+deleteAllPublisherBooks(publisherId) : void

+getSixNewestBookForBuyerDashboard() : array of array

+getSixEditorChoiceBookForBuyerDashboard() : array of array

+getSixBestsellerBookForBuyerDashboard() : array of array

+getBookForSearch(keyword) : array of array

+getLanguageName(languageId) : string

+getCategoryName(categoryId) : string

+getBookForInfoPage(id) : array

+getPeopleGaveStarsCountAllRating (bookId) : integer

+getPeopleGaveStarsCountByRating (bookId, rating) : integer

+getReviewsByBookId(bookId) : review

+getPublisherIdByBookId(bookId) : integer

+getTotalPrice(arrBookId) : integer

+getPrice(bookId) : integer

+getTitle(bookId) : string

+getBooksByArrBookId(arrBookId) : array of array

+getPublisherName(bookId) : string

+getEbookCoverId(bookId) : integer

+getSampleEbookFileId(bookId) : integer

+getEbookFileId() : array

+getBookDataForBookCollectionPage() : array of array

+getBookDataForReviewPage(bookId) : array

+isBookNotDeleted(bookId) : boolean

+getBookDataForPublisherInfoPage (publisherId) : array of array

+getBookDataForOrderInfoPage (orderId) : array of array

```
Book.php X
app > Book.php
1  <?php
2
3  namespace App;
4
5  use Illuminate\Support\Facades\DB;
6  use App\BookSnapshot;
7  use App\Publisher;
8  use App\Reviews;
9  use App\SampleEbookFile;
10 use Illuminate\Database\Eloquent\Model;
11 use Carbon\Carbon;
12
13 class Book extends Model
14 {
15     protected $table = "books";
16
17     protected $fillable = [
18         'id', 'title', 'author', 'languageId', 'numberOfPage',
19         'price', 'synopsis', 'isDeleted', 'ebookFileId',
20         'sampleEbookFileId', 'ebookCoverId', 'publisherId', 'categoryId',
21     ];
22
23     public static function store($ebookData, $ebookFilesData, $sampleEbookFilesData, $ebookCoverData)
24     {
25         DB::table('ebook_files')->insert($ebookFilesData);
26         DB::table('sample_ebook_files')->insert($sampleEbookFilesData);
27         DB::table('ebook_covers')->insert($ebookCoverData);
28         DB::table('books')->insert($ebookData);
29     }
30
31     public static function getNewBookId()
32     {
33         return DB::table('books')->get()->count() + 1;
34     }
35 }
```

```
Book.php X
app > Book.php
36     public static function getBookDataForDashboardPublisher()
37     {
38         $userId = session('id');
39         $publisherId = Publisher::getPublisherIdWithUserId($userId);
40         $books = DB::table('books')
41             ->where('publisherId', $publisherId)
42             ->where('isDeleted', '0')
43             ->get();
44         $data = [];
45         foreach ($books as $book) {
46             array_push($data, [
47                 "id" => $book->id,
48                 "title" => $book->title,
49                 "rating" => Reviews::getBookRating($book->id),
50                 "ratingCount" => Reviews::getBookRatingCount($book->id),
51                 "soldCount" => BookSnapshot::getBookSoldCount($book->id),
52                 "price" => Book::convertPriceToCurrencyFormat($book->price),
53             ]);
54         }
55         $titles = array_column($data, 'title');
56         array_multisort($titles, SORT_ASC, $data);
57         return $data;
58     }
59
60     private static function convertPriceToCurrencyFormat($price)
61     {
62         return number_format($price, 0, ',', '.');
63     }
64
65     public static function getBook($bookId)
66     {
67         return DB::table('books')
68             ->where('id', $bookId)
69             ->first();
70     }
71 }
```



```

Book.php x
app > Book.php
72     public static function updateBook($book) {
73         $data = [];
74         if ($book["title"] != null) {
75             $data = array_merge($data, array('title' => $book["title"]));
76         }
77         if ($book["author"] != null) {
78             $data = array_merge($data, array('author' => $book["author"]));
79         }
80         if ($book["languageId"] != null) {
81             $data = array_merge($data, array('languageId' => $book["languageId"]));
82         }
83         if ($book["numberOfPage"] != null) {
84             $data = array_merge($data, array('numberOfPage' => $book["numberOfPage"]));
85         }
86         if ($book["price"] != null) {
87             $data = array_merge($data, array('price' => $book["price"]));
88         }
89         if ($book["synopsis"] != null) {
90             $data = array_merge($data, array('synopsis' => $book["synopsis"]));
91         }
92         if ($book["categoryId"] != null) {
93             $data = array_merge($data, array('categoryId' => $book["categoryId"]));
94         }
95         if ($book["release_at"] != null) {
96             $data = array_merge($data, array('release_at' => $book["release_at"]));
97         }
98         $data = array_merge($data, array("updated_at" => Carbon::now()));
99         Book::where('id', $book["id"])->update($data);
100     }
101
102     public static function updateCoverPhoto($photoId, $bookId)
103     {
104         DB::table('books')
105             ->join('ebook_covers', 'books.ebookCoverId', '=', 'ebook_covers.id')
106             ->where('books.id', $bookId)
107             ->update([
108                 "ebookCoverId" => $photoId,
109                 "books.updated_at" => Carbon::now(),
110             ]);
111     }
112
113     public static function uploadSampleEbook($file, $bookId)
114     {
115         $sampleEbookId = SampleEbookFile::getNewSampleEbookFilesId();
116         $SampleEbook = $file;
117         $nama_file = $SampleEbook->getClientOriginalName();
118         $tujuan_upload = 'ebook/sample_ebook_files/'.$sampleEbookId;
119         $SampleEbook->move($tujuan_upload, $nama_file);
120         DB::table('sample_ebook_files')->insert([
121             "id" => $sampleEbookId,
122             "name" => $SampleEbook->getClientOriginalName(),
123             "created_at" => Carbon::now(),
124             "updated_at" => Carbon::now(),
125         ]);
126         DB::table('books')
127             ->join('sample_ebook_files', 'books.sampleEbookFileId', '=', 'sample_ebook_files.id')
128             ->where('books.id', $bookId)
129             ->update([
130                 "sampleEbookFileId" => $sampleEbookId,
131                 "books.updated_at" => Carbon::now(),
132             ]);
133     }
134
135     public static function deleteBook($id) {
136         $book = Book::find($id);
137         $book->isDeleted = 1;
138         $book->save();

```

```
Book.php X
app > Book.php

135     public static function deleteBook($id) {
136         $book = Book::find($id);
137         $book->isDeleted = 1;
138         $book->save();
139         Cart::removeAllBookByBookId($id);
140         Wishes::removeAllBookByBookId($id);
141     }
142
143     public static function deleteAllPublisherBooks($publisherId)
144     {
145         Cart::removeAllBookByPublisherId($publisherId);
146         Wishes::removeAllBookByPublisherId($publisherId);
147         DB::statement('UPDATE `books` SET `isDeleted` = 1 WHERE `publisherId` = '.$publisherId);
148     }
149
150     public static function getSixNewestBookForBuyerDashboard()
151     {
152         $books = DB::table('books')
153             ->select('id', 'title', 'author', 'price', 'ebookCoverId')
154             ->where('isDeleted', '0')
155             ->orderBy('release_at', 'desc')
156             ->take(6)
157             ->get();
158         foreach ($books as $book) {
159             $book->price = Book::convertPriceToCurrencyFormat($book->price);
160         }
161         $bookArray = json_decode(json_encode($books), true); //mengubah objek menjadi array
162         for ($i=0; $i < count($books); $i++) {
163             $rating = Reviews::getBookRating($bookArray[$i]["id"]);
164             $rating = number_format((float) $rating, 1, '.', '');
165             $bookArray[$i] = array_merge($bookArray[$i], array("rating" => $rating));
166             $imageUrl = EbookCover::getEbookCoverURL($bookArray[$i]["ebookCoverId"]);
167             $bookArray[$i] = array_merge($bookArray[$i], array("imageUrl" => $imageUrl));
168         }
169         return $bookArray;
170     }
171 }
```

```
Book.php X
app > Book.php
1/1

172     public static function getSixEditorChoiceBookForBuyerDashboard()
173     {
174         $books = DB::table('books')
175             ->select('id', 'title', 'author', 'price', 'ebookCoverId')
176             ->where('isDeleted', '0')
177             ->where('isEditorChoice', '1')
178             ->take(6)
179             ->get();
180         foreach ($books as $book) {
181             $book->price = Book::convertPriceToCurrencyFormat($book->price);
182         }
183         $bookArray = json_decode(json_encode($books), true); //mengubah objek menjadi array
184         for ($i=0; $i < count($books); $i++) {
185             $rating = Reviews::getBookRating($bookArray[$i]["id"]);
186             $rating = number_format((float) $rating, 1, '.', '');
187             $bookArray[$i] = array_merge($bookArray[$i], array("rating" => $rating));
188             $imageUrl = EbookCover::getEbookCoverURL($bookArray[$i]["ebookCoverId"]);
189             $bookArray[$i] = array_merge($bookArray[$i], array("imageUrl" => $imageUrl));
190         }
191         return $bookArray;
192     }
193
194     public static function getSixBestsellerBookForBuyerDashboard()
195     {
196         $booksId = DB::select("SELECT `books`.`id` FROM `books`
197             JOIN `book_snapshots` ON `books`.`id` = `book_snapshots`.`bookId`
198             JOIN `orders` ON `book_snapshots`.`orderId` = `orders`.`id`
199             WHERE `orders`.`status` = 'success'
200             AND `books`.`isDeleted` = 0
201             GROUP BY `books`.`id`
202             ORDER BY COUNT(`books`.`id`) DESC
203             LIMIT 6"); //berupa array of object (array biasa)
204         if (count($booksId) == 0) { // Jika tidak ada buku yang terjual
205             return null;
206         }
207         $arrBooksId = [];
208         array_splice($arrBooksId, 0, 1, $booksId);
209     }
```

```

Book.php
app > Book.php
206     }
207     $arrBooksId = [];
208     array_splice($arrBooksId,0,1,$booksId);
209     $pureArrBooksId = [];
210     for ($i=0; $i < count($arrBooksId); $i++) {
211         if ($i == 0) {
212             $pureArrBooksId[0] = $arrBooksId[0]->id;
213         }
214         else {
215             array_push($pureArrBooksId, $arrBooksId[$i]->id);
216         }
217     }
218     $books = DB::table('books')
219         ->select('id', 'title', 'author', 'price', 'ebookCoverId')
220         ->where('isDeleted', '0')
221         ->whereIn('id', $pureArrBooksId)
222         ->take(6)
223         ->get();
224     foreach ($books as $book) {
225         $book->price = Book::convertPriceToCurrencyFormat($book->price);
226     }
227     $bookArray = json_decode(json_encode($books), true); //mengubah objek menjadi array
228     for ($i=0; $i < count($books); $i++) {
229         $rating = Reviews::getBookRating($bookArray[$i]["id"]);
230         $rating = number_format((float) $rating, 1, '.', '');
231         $bookArray[$i] = array_merge($bookArray[$i], array("rating" => $rating));
232         $imageUrl = EbookCover::getEbookCoverURL($bookArray[$i]["ebookCoverId"]);
233         $bookArray[$i] = array_merge($bookArray[$i], array("imageUrl" => $imageUrl));
234         $soldCount = BookSnapshot::getBookSoldCount($bookArray[$i]["id"]);
235         $bookArray[$i] = array_merge($bookArray[$i], array("soldCount" => $soldCount));
236     }
237     \usort($bookArray, function ($book1, $book2) {
238         return $book2['soldCount'] <=> $book1['soldCount'];
239     });
240     return $bookArray;
241 }

```

```

Book.php X
app > Book.php
242
243 public static function getBookForSearch($keyword)
244 {
245     $keywords = explode(" ", $keyword);
246     $addOnQuery = " ";
247     foreach ($keywords as $kataKunci) {
248         $addOnQuery = $addOnQuery."AND `title` LIKE '%" . $kataKunci . "%' ";
249     }
250     $query = "SELECT
251         `books`.`id`, `title`, `author`, `languageId`, `price`, `release_at`, `synopsis`,
252         `ebookCoverId`, `categoryId`,
253         `languages`.`name` as 'Language',
254         `ebook_covers`.`name` as 'ebookCoverName',
255         `categories`.`name` as 'Category'
256     FROM `books`
257     JOIN `ebook_covers` ON `books`.`ebookCoverId` = `ebook_covers`.`id`
258     JOIN `languages` ON `books`.`languageId` = `languages`.`id`
259     JOIN `categories` ON `books`.`categoryId` = `categories`.`id`
260     WHERE `isDeleted` = 0";
261     $query = $query.$addOnQuery;
262     $books = DB::select($query, [1]);
263     foreach ($books as $book) {
264         $book->rating = Reviews::getBookRating($book->id);
265         $book->ratingCount = Reviews::getBookRatingCount($book->id);
266         $book->soldCount = BookSnapshot::getBookSoldCount($book->id);
267         $book->priceForHuman = Book::convertPriceToCurrencyFormat($book->price);
268     }
269     return response()->json($books);
270 }
271
272 private static function getLanguageName($languageId)
273 {
274     return DB::table('languages')
275         ->where('id', $languageId)
276         ->pluck('name')[0];
277 }
278

```

```

Book.php X
app > Book.php
279 private static function getCategoryName($categoryId) {
280     return DB::table('categories')
281         ->where('id', $categoryId)
282         ->pluck('name')[0];
283 }
284
285 public static function getBookForInfoPage($id)
286 {
287     $book = DB::table('books')
288         ->where('id', $id)
289         ->first();
290     $book->priceForHuman = Book::convertPriceToCurrencyFormat($book->price);
291     $book->language = Book::getLanguageName($book->languageId);
292     $book->category = Book::getCategoryName($book->categoryId);
293     $book->publisher = Publisher::getPublisherName($book->publisherId);
294     $rating = Reviews::getBookRating($book->id);
295     $rating = number_format((float) $rating, 1, '.', '');
296     $book->rating = $rating;
297     $book->ratingCount = Reviews::getBookRatingCount($book->id);
298     $book->soldCount = BookSnapshot::getBookSoldCount($book->id);
299     $book->imageUrl = EbookCover::getEbookCoverURL($book->ebookCoverId);
300     unset($book->ebookCoverId);
301     unset($book->created_at);
302     unset($book->updated_at);
303     unset($book->price);
304     unset($book->isDeleted);
305     unset($book->isEditorChoice);
306     return json_decode(json_encode($book), true);
307 }
308
309 public static function getPeopleGaveStarsCountAllRating($bookId)
310 {
311     return DB::table('reviews')
312         ->join('have', 'reviews.haveId', '=', 'have.id')
313         ->where('have.bookId', $bookId)
314         ->count();
315 }
316
317 public static function getPeopleGaveStarsCountByRating($bookId, $rating)
318 {
319     return DB::table('reviews')
320         ->join('have', 'reviews.haveId', '=', 'have.id')
321         ->where('have.bookId', $bookId)
322         ->where('rating', $rating)
323         ->count();
324 }
325
326 public static function getReviewsByBookId($bookId)
327 {
328     $reviews = DB::table('reviews')
329         ->select('reviews.id', 'rating', 'review', 'isAnonymous', 'firstName', 'lastName', 'isDeleted', 'reviews.created_at')
330         ->join('have', 'reviews.haveId', '=', 'have.id')
331         ->join('users', 'have.userId', '=', 'users.id')
332         ->where('have.bookId', $bookId)
333         ->orderBy('reviews.id', 'asc')
334         ->get();
335     foreach ($reviews as $review) {
336         $created_at = Carbon::parse($review->created_at)->toDateString();
337         $review->created_at = $created_at;
338     }
339     return $reviews;
340 }
341
342 public static function getPublisherIdByBookId($bookId)
343 {
344     return DB::table('books')
345         ->where('id', $bookId)
346         ->get()[0]->publisherId;
347 }
348
349 public static function getTotalPrice($arrBookId)
350 {
351     $totalPrice = 0;
352     foreach ($arrBookId as $book) {
353         $totalPrice = $totalPrice + DB::table('books')->where('id', $book->bookId)->pluck('price')[0];

```

```

349 public static function getTotalPrice($arrBookId)
350 {
351     $totalPrice = 0;
352     foreach ($arrBookId as $book) {
353         $totalPrice = $totalPrice + DB::table('books')->where('id', $book->bookId)->pluck('price')[0];
354     }
355     return $totalPrice;
356 }
357
358 public static function getPrice($bookId)
359 {
360     return DB::table('books')->where('id', $bookId)->pluck('price')[0];
361 }
362
363 public static function getTitle($bookId)
364 {
365     return DB::table('books')->where('id', $bookId)->pluck('title')[0];
366 }
367
368 public static function getBooksByArrBookId($arrBookId)
369 {
370     return DB::table('books')->whereIn('id', $arrBookId)->get();
371 }
372
373 public static function getPublisherName($bookId)
374 {
375     $publisherId = DB::table('books')->where('id', $bookId)->pluck('publisherId')[0];
376     return Publisher::getPublisherName($publisherId);
377 }
378
379 public static function getEbookCoverId($bookId)
380 {
381     return DB::table('books')->where('id', $bookId)->pluck('ebookCoverId')[0];
382 }
383

```

```

384 public static function getSampleEbookFileId($bookId)
385 {
386     return DB::table('books')->where('id', $bookId)->pluck('sampleEbookFileId')[0];
387 }
388
389 public static function getEbookFileId($bookId)
390 {
391     return DB::table('books')->where('id', $bookId)->pluck('ebookFileId')[0];
392 }
393
394 public static function getBookDataForBookCollectionPage()
395 {
396     $books = DB::table('books')
397         ->join('have', 'have.bookId', '=', 'books.id')
398         ->join('ebook_covers', 'books.ebookCoverId', '=', 'ebook_covers.id')
399         ->where('have.userId', '=', session('id'))
400         ->select('title', 'author')
401         ->addSelect(DB::raw("`books`.`id` as id"))
402         ->addSelect(DB::raw("`ebook_covers`.`id` as ebookCoverId"))
403         ->addSelect(DB::raw("`ebook_covers`.`name` as ebookCoverFileName"))
404         ->get();
405     foreach ($books as $book) {
406         $book->didTheUserGiveAREview = Reviews::didTheUserGiveAREview($book->id);
407     }
408     return $books;
409 }
410
411 public static function getBookDataForReviewPage($bookId)
412 {
413     $book = DB::table('books')
414         ->join('publishers', 'publishers.id', '=', 'books.publisherId')
415         ->join('ebook_covers', 'books.ebookCoverId', '=', 'ebook_covers.id')
416         ->where('books.id', $bookId)
417         ->select('books.id', 'title', 'author')
418         ->addSelect(DB::raw("`publishers`.`name` as publisherName"))
419         ->addSelect(DB::raw("`ebook_covers`.`id` as ebookCoverId"))
420         ->first();

```

```

421     $book->ebookCoverURL = EbookCover::getEbookCoverURL($book->ebookCoverId);
422     unset($book->getEbookCoverId);
423     return $book;
424 }
425
426 public static function isBookNotDeleted($bookId)
427 {
428     $count = DB::table('books')
429         ->where('id', $bookId)
430         ->where('isDeleted', 0)
431         ->get()
432         ->count();
433     return $count == 1;
434 }
435
436 public static function getBookDataForPublisherInfoPage($publisherId)
437 {
438     $books = DB::table('books')
439         ->where('publisherId', $publisherId)
440         ->where('isDeleted', '0')
441         ->get();
442     $data = [];
443     foreach ($books as $book) {
444         $rating = Reviews::getBookRating($book->id);
445         $rating = number_format((float) $rating, 1, '.', '');
446         array_push($data,[
447             "id" => $book->id,
448             "title" => $book->title,
449             "author" => $book->author,
450             "imageURL" => EbookCover::getEbookCoverURL($book->ebookCoverId),
451             "synopsis" => $book->synopsis,
452             "rating" => $rating,
453             "ratingCount" => Reviews::getBookRatingCount($book->id),
454             "soldCount" => BookSnapshot::getBookSoldCount($book->id),
455             "price" => Book::convertPriceToCurrencyFormat($book->price),
456         ]);
457     }

```

Book.php X

app > Book.php

```

456     ]);
457 }
458 return $data;
459 }
460
461 public static function getBookDataForOrderInfoPage($orderId)
462 {
463     $arrBookId = BookSnapshot::getArrBookIdByOrderId($orderId);
464     $books = DB::table('books')
465         ->whereIn('id', $arrBookId)
466         ->select('id', 'title', 'author', 'publisherId', 'ebookCoverId')
467         ->get();
468     foreach ($books as $book) {
469         $price = BookSnapshot::getPrice($book->id, $orderId);
470         $book->price = Book::convertPriceToCurrencyFormat($price);
471         $book->publisherName = Publisher::getPublisherName($book->publisherId);
472         $book->coverURL = EbookCover::getEbookCoverURL($book->ebookCoverId);
473         unset($book->publisherId);
474         unset($book->ebookCoverId);
475     }
476     return $books;
477 }
478 }
479

```

4. Screenshoot bukti pengisian form feedback IFLAB <http://bit.ly/FeedbackPraUTS>

- Naufal



The screenshot shows a Google Form interface. At the top, there is a logo for 'informatics lab' consisting of a stylized 'if' in blue with purple curly braces on either side. Below the logo, the title 'Feedback Terhadap Pelayanan Informatics Lab' is displayed in a large, bold, black font. Underneath the title, a smaller line of text reads 'Terima kasih atas feedback yang telah Anda berikan.' At the bottom of the form, there is a small text that says 'This form was created inside of Telkom University. Report Abuse' and the 'Google Forms' logo.

- Saskia



This screenshot is identical to the one above, showing the same Google Form titled 'Feedback Terhadap Pelayanan Informatics Lab'. It features the same 'informatics lab' logo, title, and thank-you message. The bottom text 'This form was created inside of Telkom University. Report Abuse' and the 'Google Forms' logo are also present.

- Anis



informatics lab

Feedback Terhadap Pelayanan Informatics Lab

Terima kasih atas feedback yang telah Anda berikan.

This form was created inside of Telkom University. [Report Abuse](#)

Google Forms

- Ingrid



You've already responded

You can fill out this form only once.

Try contacting the owner of the form if you think this is a mistake.

This form was created inside of Telkom University. [Report Abuse](#)

Google Forms

#Selamat bekerja

RUBRIKASI

- A : jawaban lengkap soal 1, 2, 3a + 3b, 3c
- AB : jawaban soal 1 lengkap, 2, 3a + 3b, sebagian 3c
- B : jawaban soal 1 sebagian, 2, 3a + 3b, sebagian 3c
- BC : jawaban soal 2, 3a + 3b, sebagian 3c
- C : jawaban soal 2, 3a + 3b
- D : jawaban soal 2 & 3a
- E : tidak hadir