



Web-Based College Student Assignment File Collection Application Using Google Drive API

Khairi Ibnutama*, Hendryan Winata, Masyuni Hutasuht

Department of Information System, STMIK Triguna Dharma, North Sumatera, Indonesia

Email: ^{1*}mr.ibnutama@gmail.com

Abstract—The use of internet media, especially cloud storage as a practical solution in information and communication should be able to be applied in academics, especially in the scope of teaching and learning. By using cloud storage as a communication medium, especially in sending data in the form of assignments files can facilitate lecturers and students in carrying out one of the mandatory activities in lectures, namely the collection of assignments. The main problem that often occurs is the storage capacity provided by hosting services is very limited, so storing large amounts of files will require large costs as well. Google Drive provides an API driven file storage service that makes it easy for users to create applications that can communicate and interact with 15 GB free storage services. The free service is expected to facilitate teaching and learning activities in lectures, especially in the collection of student assignments.

Keywords: API, Google Drive, Academics, Website, Application

1. INTRODUCTION

STMIK Triguna Dharma is an educational institution where lecturers who teach are required to give assignments to students in one semester. The value of the assignment is one of several data values inputted on the student grade form. The problem of time is the main factor of the difficulty of students to collect assignments, where students must meet the lecturer concerned to submit the results of the work done. Matching the schedule of meetings between students and lecturers becomes difficult because of the tight schedule of the lecturers concerned. But if internet media can be optimized for use, of course time is not a problem in gathering student assignments. For this reason, this research is expected to make it easier for students to collect assignment files, as well as facilitate lecturers in managing the assignments list for each class.

Utilization of online media as a means of collecting files is certainly constrained by the limitations of hosting server capacity. Therefore, by using the Google Drive as a cloud storage, it is expected to be a solution as a media choice for storing assignment files collected by students. Google, which is a provider of services and internet products, provides free file storage media services with a capacity of 15 GB under the name Google Drive[1]. These services can be used as a substitute for limited storage provided by paid hosting. Also Google Drive does not limit the types of files that can be uploaded to the server as long as the file is not detected by viruses and malware.

Google Drive provides Application Programming Interface (API) facilities that allow users to create applications that can communicate and interact with these services[2]. The API facility enables the creation of a student assignment file collection website application where each file uploaded by a student will automatically be saved in the lecturer's Google Drive account. Utilization of the Google Drive API is expected to be an easy and practical solution for lecturers and students in carrying out teaching and learning activities.

Application interface is made using the PHP programming language and JavaScript on web pages. Student data in the form of ID number, student name, and class name will be stored on the server database, while assignment files uploaded on the website will be transferred and stored to Google Drive using the API. After that the url file address obtained will be stored in the server database.

2. THEORY

2.1 PHP MySQL

PHP or Hypertext Preprocessor is a server-side programming language created specifically for websites. The program code in PHP will be interpreted on the web server and displayed in HTML or other output that can be seen by visitors[3]. In this research, the PHP CodeIgniter framework uses the PHP MVC (Model-View-Controller) system for server-side scripting.

PHP is designed to be able to accelerate the construction process of a website, and is enriched with many libraries that can interact with databases. One of the most commonly used libraries in a database is the original MySQL API library[4]. PHP is often paired with MySQL as an application database as shown in Fig. 1. Special



commands in PHP linking web pages with MySQL database to be sent to web pages in HTML form in response to data requests [3].

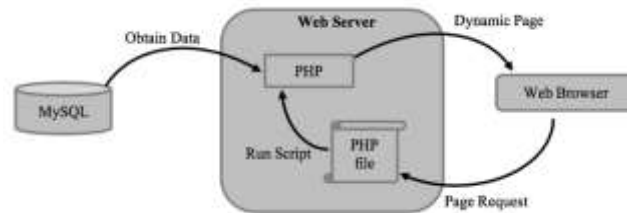


Figure 1. PHP and MySQL Data Communication

In other studies[5] also explained that, one possible way to simplify the website development process for developers without having to create an intermediary agent or custom service is to create a "blackbox" that takes input in the form of SQL statements that will communicate with the database and display the data becomes various representations such as interactive tables or graphs.

2.2 Google Drive API

Google Drive is a storage medium in the form of cloud applications made by Google that can be integrated with many other Google applications. Google Drive was launched on April 24, 2012 which is an update of Google Docs with a capacity of up to 15GB including Gmail and Google Photos[1]. Google Drive services can be accessed and used with only one registered Google account, either through the web platform, Android or iOS[6].

APIs are rules that can be used to access a software or application service. The API functions as a software interface or application service so that it can be accessed by other software or applications. API services can be analogous to interactions between humans and computers[6][7].

The Google Drive Software Development Kit (SDK) includes an HTTP API that lets users or developers integrate, modify, and access the files stored in a user's Drive with their own third party applications. This gives users and app developers the ability to use multiple cloud apps to access and interact with their files that are stored in a single location in the cloud. Other studies[8] also mention the use of Google Api for PHP-based web interoperability models with Google Drive. This study discusses the use of Google Drive Api in supporting the interoperability model of the existing system using the UML (Unified Modeling Language) tool and the Google Drive API as its implementation system. Daryabar[9] also mentioned in his research that metadata files uploaded on Google Drive can be used as forensic investigation material. Thus a lot of information related to files can be saved and used if needed later.

3. RESULT AND DISCUSSION

Web applications that will be made are divided into two stages, namely system planning and API communication based on theories that has been mentioned before.

3.1 System Planning

Planning is the initial stage in making an information system consisting of a database and application interface to make the best user experience as possible.

3.1.1 Database Design

In accordance with the needs of the system to be created, a database design is carried out containing tables that serve to store information of application and students who collect assignments.

Table 1. Tables in the kumpultugas Database

No.	Table Name
1	userinfo
2	tugasinfo

a. Userinfo Table

Database : kumpultugas

Table Name : userinfo



Primary Key : id

Description : To store user (lecturer) information along with application settings information.

Table 2. Userinfo Table

Name	Type	Extra	Action
id	int(8)	AUTO_INCREMENT	Primary
username	varchar(100)	-	-
password	varchar(255)	-	-
email	varchar(100)	-	Unique
fullname	varchar(100)	-	-
scripturl	text	-	-

b. Tugasinfo Table

Database : kumpultugas

Table Name : tugasinfo

Primary Key : id

Description : To store student assignments file information.

Table 3. Tugasinfo Table

Name	Type	Extra	Action
id	int(8)	AUTO_INCREMENT	Primary
dosen_id	int(8)	-	Foreign Key
nim	varchar(12)	-	-
std_name	varchar(100)	-	-
class_name	varchar(100)	-	-
session_asg	int(2)	-	-
uploaded_at	timestamp	-	-
fileurl	text	-	-

3.1.2 Interface Design

At this stage the application interface is designed from the front-end, which is the page/section accessed by students to collect assignments, and the back-end as the lecturer page/section.

a. Front-end Application

Front-end application is a page that contains a form of filling student data that will upload the assignment file. The form uses the attribute enctype="multipart/form-data" as an HTML facility to handle HTTP request for uploading files on web pages[10].

Figure 2. Front-end Form Application Design

b. Back-end Application

The back-end application is a page that can only be accessed by lecturers containing information on student assignment files that have been collected on the website along with the assignments file link stored on Google Drive. This page contains CRUD operation for managing that information[4][5] that displayed in the form of a table[5].



Pilih Kelas ▼		Pilih Sesi Tugas ▼		SEARCH	
No.	NIM	Nama Mahasiswa	Tgl. Upload	Jam	OPTIONS
1	-----	-----	XX/XX/XXXX	XX:XX	Download Delete

Figure 3. Back-end Application Interface

3.2 API Communication

Communication between the web application and the Google Drive API is done on the form contained in the application front-end based on application flowchart that can be seen in Fig 4.

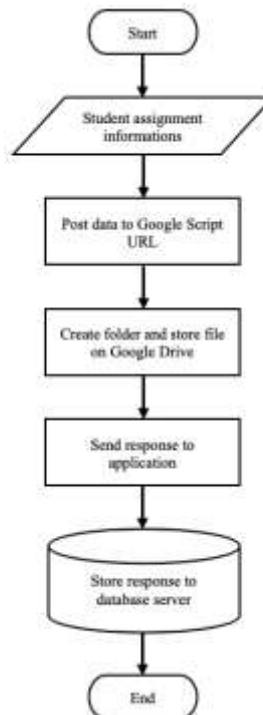


Figure 4. Application Flowchart

Google Drive API using Google Script has two functions that can be used for collecting information sent from another app. When a user visits an application sends the script an HTTP GET request, Google Script runs the function doGet(e). When an application sends the script an HTTP POST request, Google Script runs doPost(e) instead. The e argument represents an event parameter that can contain information sent from application about any request parameters in both cases.



Figure 5. Google Script Argument

Google Script can be created and deployed from Google Drive "New" options on <https://drive.google.com> after the user login using their Google Account. The deployed script result a script URL used for any other third party applications to interact with Google Drive that linked to the script.



4. IMPLEMENTATION

The steps for implementing web application communication with Google Drive are based on the following chart.

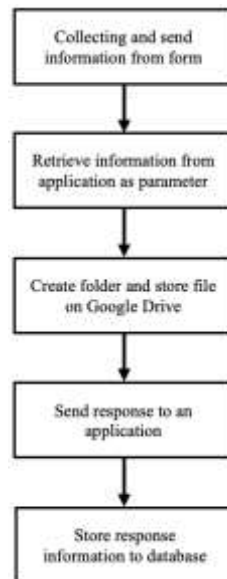


Figure 6. Application Workflow

4.1 Collecting and Send Information

Information provided by student on front-end application are collected using JavaScript and submitted to script URL using POST method.

```
function UploadFile(){
  var reader = new FileReader();
  var file = document.getElementById('filetugas').files[0];
  reader.onload = function(){
    document.getElementById('fileContent').value=reader.result;
    document.getElementById('fileName').value=file.name;
    document.getElementById('formName').submit();
  }
  reader.readAsDataURL(file);
}
```

4.2 Retrieve Information on Google Script

Informations that sent from application to Google Script are retrieved using doPost(e) function to be managed as folder and file name.

```
function doPost(e){
  var data      = e.parameter.fileContent;
  var fname     = e.parameter.fileName;
  var ext       = fname.substring(fname.lastIndexOf('.')+1, fname.length) || fname;
  var nim       = e.parameter.nim;
  var stdName   = e.parameter.stdName;
  var clsName   = e.parameter.clsName;
  var sesAsg    = 'TUGAS' + e.parameter.sessionAsg;
  var upload    = uploadToGdrive(data,fname,ext,nim,stdName,clsName,sesAsg,e);
}
```

4.3 Create Folder and Storing File to Google Drive

Informations that has been retrieved are used to create folder on Google Drive and rename the file based on information provided.

```
function uploadToGdrive(data,fname,ext,nim,stdName,clsName,sesAsg,e){
```



```
var parent      = DriveApp.getFoldersByName('FILES');
if(parent.hasNext()){
  parent.next();
  createFolder(sesAsg);
  createFolder(clsName);
  var file      = createFile(blob);
  var fUrl     = file.getUrl();
  return fUrl;
}
}
```

4.4 Send Response to Application

After file successfully uploaded to Google Drive, a line of response is sent as a feedback from Google Script to application. The response determine wether student information will be saved to database or not.

```
if(fUrl){
  var resp      = "<script>>window.top.location.href='"+SERVER+"/savetugas/?nim="
                +nim+"&stdName="+name+"&cls="+clsName+"&ses="+sesAsg+"&url="
                +fUrl+"";</script>";
}
return HtmlService.createHtmlOutput(resp);
```

4.5 Store Response to Database

Response sent from Google Script are retrieved on web application contain information related to file that has been submitted. The information will be processed using PHP GET method and store to database.

```
public function savetugas (){
  $data = array(
    'nim' => $this->input->get('nim'),
    'std_name' => $this->input->get('stdName'),
    'class_name' => $this->input->get('cls'),
    'session_asg' => $this->input->get('ses'),
    'fileurl' => $this->input->get('url')
  );
}
```

Informations that has been stored will be accessed again on the back-end page for further management.

5. CONCLUSION

Collection of student assignment files using Google Drive as a file storage medium allows to overcome the problem of capacity limitations provided by paid or free hosting. In addition, by utilizing Google's cloud computing and Firebase services, this research can be further developed by adding a real-time notification feature every time a student collects an assignment.

REFERENCES

- [1] R. Cahaya, I. Made, and A. A., "Data Exchange Service using Google Drive API", *Int. Jml. of Comp. App.*, vol. 154, no. 7, pp. 12-16, 2016.
- [2] S. Challita, F. Zalila, C. Gourdin, and P. Merle, "A Precise Model for Google Cloud Platform", *IC2E*, pp. 177-183, 2018.
- [3] Hardono, I. Sujandari, A. Rachman, Y. Panjaitan, and A. Rosdiyah, "Development of Theses Categorization System Search Engine using PHP and MySQL", *ICITSI*, pp. 194-199, 2017.
- [4] D. Anderson and M. Hills, "Query Construction Patterns in PHP", *SANER*, pp. 452-456, 2017.
- [5] E. Hua, V. Nen, F. Tee, and O. Ann, "Pigeon-Chart: A Customized HTML Element for Data Visualization in Data-Driven Web Application Using AngularJS, HighCharts, UnderscoreJS and PHP", *ICCIS*, pp. 247-252, 2018.
- [6] V. Raymond and E. Sushmitha, "Google Drive Based Secured Anti-Theft Android Application", *ICIOT*, pp. 1-8, 2017.
- [7] W. Li, C. Yen, Y. Lin, S. Tung, and S. Huang, "JustIoT Internet of Things based on the Firebase real-time database", *SMILE*, pp. 43-47, 2018.
- [8] U. Rahardja, Q. Aini, and N. Santoso, "Pengintegrasian Yii Framework Berbasis API pada Sistem Penilaian Absensi", *SISFOTENIKA*, vol. 8, no. 2, pp. 140-152, 2018.



- [9] F. Daryabar, A. Dehghantanha, B. Eterovic-Soric, and K. Choo, “Forensic Investigation of OneDrive, Box, GoogleDrive and Dropbox Applications on Android and iOS Devices”, *Australian Jnl. of Forensic Sciences*, vol. 48, no. 6, pp. 615-642, 2016.
- [10] A. Fink, “A Mobile Application for Crowdsourced Acquisition of Urban Street-View Pedestrian Facility Data”, *Honors Theses*, vol. 5, no. 654, pp. 1-39, 2019.