

Implementation of Google Cloud in Business to Business *Tanda Tukar Faktur* Application

Benny Kurniawan¹, Radius Tanone²

^{1,2}Information Technology Faculty, Universitas Kristen Satya Wacana, Indonesia

¹bernardus.707@gmail.com, ²radius.tanone@uksw.edu

Abstract - Alfamart is a company engaged in retail. Companies involved in the retail sector are certainly inseparable from buying and selling products, and every transaction that occurs will be detailed in the invoice exchange. The problem that arises is because Alfamart wants to accommodate the electronic invoice exchange process. Therefore, Alfamart built a B2B TTF application that can accommodate the electronic invoice exchange process and help its accounting management. The application is made using the Research and Development method because it can address urgent needs and has a high validation value. It is built using the Flask framework and is integrated with Google Cloud to overcome application deployment speed problems and be more flexible. In addition, the implementation of Optical Character Recognition using Google Vision is used to validate uploaded invoice files. This study's results are in the form of a B2B TTF application that can make it easier for users to exchange invoices. The results of using Google Vision have a relatively high percentage of 77%. The B2B TTF application uses the Flask framework and is integrated with Google Cloud, which can assist users in the process of exchanging invoices electronically.

Keywords: exchange invoices, *flask*, *Google Cloud*

I. INTRODUCTION

Tanda Tukar Faktur (TTF) is a commercial document that details transactions between sellers and buyers. Also, TTF functions as an internal control in a company's accounting. PT. Sumber Alfaria Trijaya, Tbk or better known as Alfamart, is one of several large companies engaged in retail and has more than 14,300 outlets spread across Indonesia. Every company engaged in the retail sector is undoubtedly inseparable from buying and selling products, and every transaction that occurs will be detailed in the TTF. Currently, the TTF input process at Alfamart is still carried out face-to-face, so there are obstacles faced, such as limited time to input TTF, which causes some invoice exchange processes to be delayed. Therefore, an application that can accommodate the TTF input process electronically and help its accounting management is needed. The Business to Business *Tanda*

Tukar Faktur (B2B TTF) application was built to solve problems.

B2B TTF applications are built using the Flask framework because Flask gives developers the freedom to choose which components to use so that unused parts do not burden applications built. Also, Flask provides a Web Server Gateway Interface (WSGI) toolkit and the Jinja2 template engine, which functions to create templates for the user interface [1-2]. The Bootstrap framework is also used to build the web with the Responsive Web Design (RWD) method. The B2B TTF application implements the Single Page Application (SPA) using Ajax technology to make it easier for users to input TTF and build user-friendly applications.

Because it has a large scale of use, an idea to integrate with Google Cloud was put forward to make it more flexible and secure data security [3] and to implement the Flask framework so that the memory capacity used for applications is not too heavy so that it speeds up the application deployment process [4]. Some of the Google Cloud services used to build B2B TTF applications include Google App Engine, Google Compute Engine, Google Cloud Storage and Google Vision. Utilizing the Google App Engine's existing features in prototype deploying makes it easier for employees to test applications. The deployment of the B2B TTF application is carried out to the Google App Engine so that employees can access it without having to be connected to one network. Also, Google Compute Engine is used to make API requests to the e-invoice web, useful for invoice validation and getting invoice response data. The invoice file will be stored in Google Cloud Storage. At the same time, Google Vision will be used to carry out Optical Character Recognition (OCR) on the invoice file to detect the transaction's total value. The OCR process on the invoice file is carried out to avoid the invoice file being uploaded in a dummy file or empty file.

Based on the background above, in this study, a web-based application design and integrated with Google Cloud are carried out to make it more flexible and secure data security, and implement the Flask framework so that

the capacity used for applications is not too heavy, thereby speeding up the application deployment process. By doing this research, it is hoped that it will help Alfamart employees in managing purchase transaction data with suppliers and the B2B TTF application also provides electronic sending of invoice files to make it easier for employees and suppliers in a more flexible invoice exchange process.

Research that is similar to this research supports and can be used as a reference for writing research including, The research conducted to provide a study of the benefits of implementing Google Cloud for companies to increase business competitiveness. This research also aims to provide an overview of cloud computing's meaning and its development and the advantages and disadvantages of companies using cloud computing technology. Based on this research, it is known that cloud computing is used as a technology that utilizes internet services using a virtual server centre to maintain data and applications [3].

Another research reference aimed at discussing the application of project time management, which is one of the applications used in groups and serves as a medium for project monitoring and control. Determining the schedule, implementation progress, solving problems that occur and completing the project on time are the study's objectives. The resulting application is designed using Google Cloud Platform (GCP). Project time management applications have a dependence on real-time time, so they require high system specifications, and the necessary costs are expensive. One solution to overcome this research problem is to build an application with a cloud computing model that utilizes the service provider's infrastructure capabilities to produce scalable and fault-tolerant services [5].

There is also a research reference regarding how cloud computing is applied in the education sector and the benefits of cloud computing. This study aims to determine whether cloud computing can be used in the education sector and the services provided by cloud computing. The results of this study have benefits, such as increasing efficiency and effectiveness in the education sector [6].

From the journal literature above, the idea of implementing Google Cloud Service in the Business to Business Tanda Tukar Faktur Application was created. The application of cloud computing in an application has benefits such as minimizing costs for infrastructure, making it easier for data to be accessed only by utilizing internet services and cloud service providers providing storage according to user requests, making it easier for users to manage data to be stored on cloud servers. This

method has proven to accommodate the electronic invoice exchange process, proven to accelerate application integration and implementation.

II. METHOD

Research and Development (R&D) used in the research method, which is then adjusted to the research objectives and applied in research cases. The research stages can be described as in Fig. 1.

A. Potential and Problems

At this stage, the identification of problems at PT. Sumber Alfaria Trijaya, Tbk, in this case, is a submission of an application request from employees of PT. Sumber Alfaria Trijaya, Tbk, also acts as a user. The proposal submitted is that the user wants to accommodate sending invoice files electronically. From this request, a B2B TTF application was designed to accommodate the sending of invoice files electronically using e-stamp duty.

B. Data Collection

The necessary data is collected to conduct research. The data collected was obtained from PT. Sumber Alfaria Trijaya, Tbk in the Information Technology department of the Business Intelligence division. The data required include employees data, branch office data, supplier data, billing data, invoice data, and *Goods Purchase List* (LPB) data.

C. Product Design

The system design for the B2B TTF application is carried out at this stage. Design system designed by employees of PT. Sumber Alfaria Trijaya, Tbk department of Information Technology, Demand division. The system design is made in a flowchart and a user interface with application details. Fig. 2 is a B2B

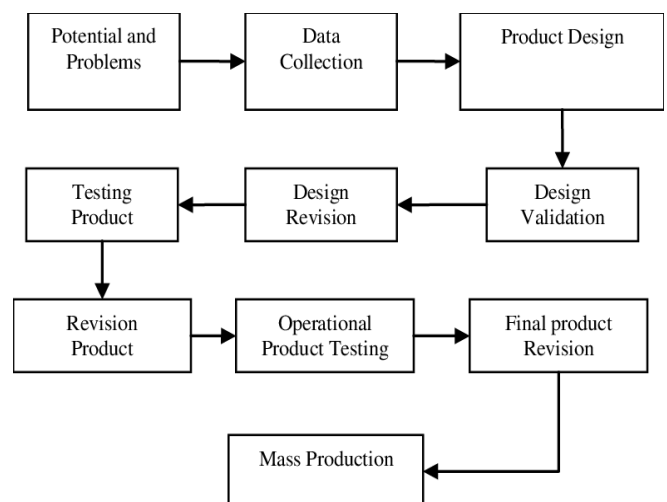


Fig. 1 R&D research stages

TTF application flowchart that explains an overview of the application's running. Starting from the user, the user selects the branch code and supplier code. The system will make a draft. After the draft number is formed, the system will check the user status. Suppose the user is *Pengusaha Kena Pajak* (PKP). In that case, the user can input invoices, LPB and bills, whereas if the user status is *Non-Pengusaha Kena Pajak* (NPKP), the user can only input LPB and bills.

D. Design Validation

This stage carried out an assessment of the design system that the employees of PT. Sumber Alfaria Trijaya, Tbk department of Information Technology, Demand division have designed, by presenting the application in prototype and discussing with users about the effectiveness of application design.

E. Design Revision

The employees of PT. Sumber Alfaria Trijaya, Tbk, the Information Technology department of the Demand division made changes to a design system that was less than optimal or that was not by user requests at this stage.

F. Testing Product

The application is tested to look for bugs and ensure that the application is run following the design system. This trial phase was carried out by the PT. Sumber Alfaria Trijaya, Tbk department of Information Technology employees, Quality Assurance Division.

G. Revision Product

At this stage, PT. Sumber Alfaria Trijaya, Tbk employees of the Information Technology department of the Quality Assurance division provide feedback to the developer to make bug fixes that occur in the application.

H. Operational Product Testing

The application trial is carried out to the user concerned regarding the application built by what was proposed. In this research, the user concerned is an employee of PT. Sumber Alfaria Trijaya, Tbk in the finance division.

I. Final Product Revision

At this stage, the user gives the developer feedback to make final improvements before the application is launched to the public. The developer ensures that the application can run by the design system without any problems.

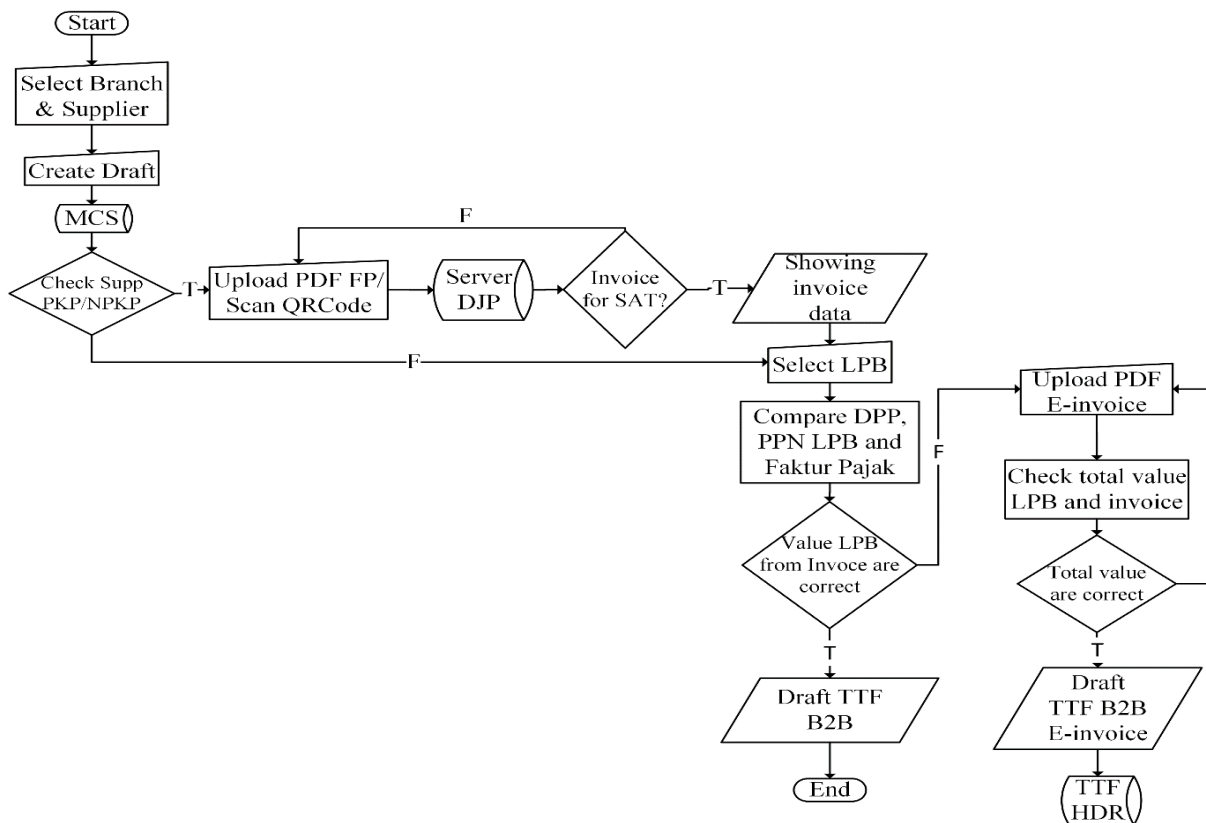


Fig. 2 Flowchart B2B TTF application

J. Mass Production

The application is launched at this stage and employees of PT. Sumber Alfaria Trijaya, Tbk, the Merchandiser Display division, provides direction and training to suppliers in using this B2B TTF application [7].

III. RESULTS AND DISCUSSION

This research resulted in a web-based B2B TTF application built using the Python programming language and integrated with Google Cloud and implementing the Flask framework as a web server. There are technologies such as PostgreSQL, which act as databases and Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Bootstrap, Javascript, and Jinja2 Templates to build user interfaces. In Fig. 3, you

can see the user interface that has been built using Python technology and the Flask framework.

The Flask framework is implemented because Flask allows developers to choose libraries to use to avoid being overwhelmed by unnecessary libraries. The Flask project structure applies the MVC design pattern, and the structure can be seen in Fig. 4. The controller is used to accommodate functions in charge of running each process in the application, while the DAO will be used as a container for tasks that play a role in the data processing. Models contain modelling data from the application, assets supporting files such as images and icons and static, which will accommodate the Bootstrap and JavaScript libraries and templates that play a role in displaying the user interface [8].

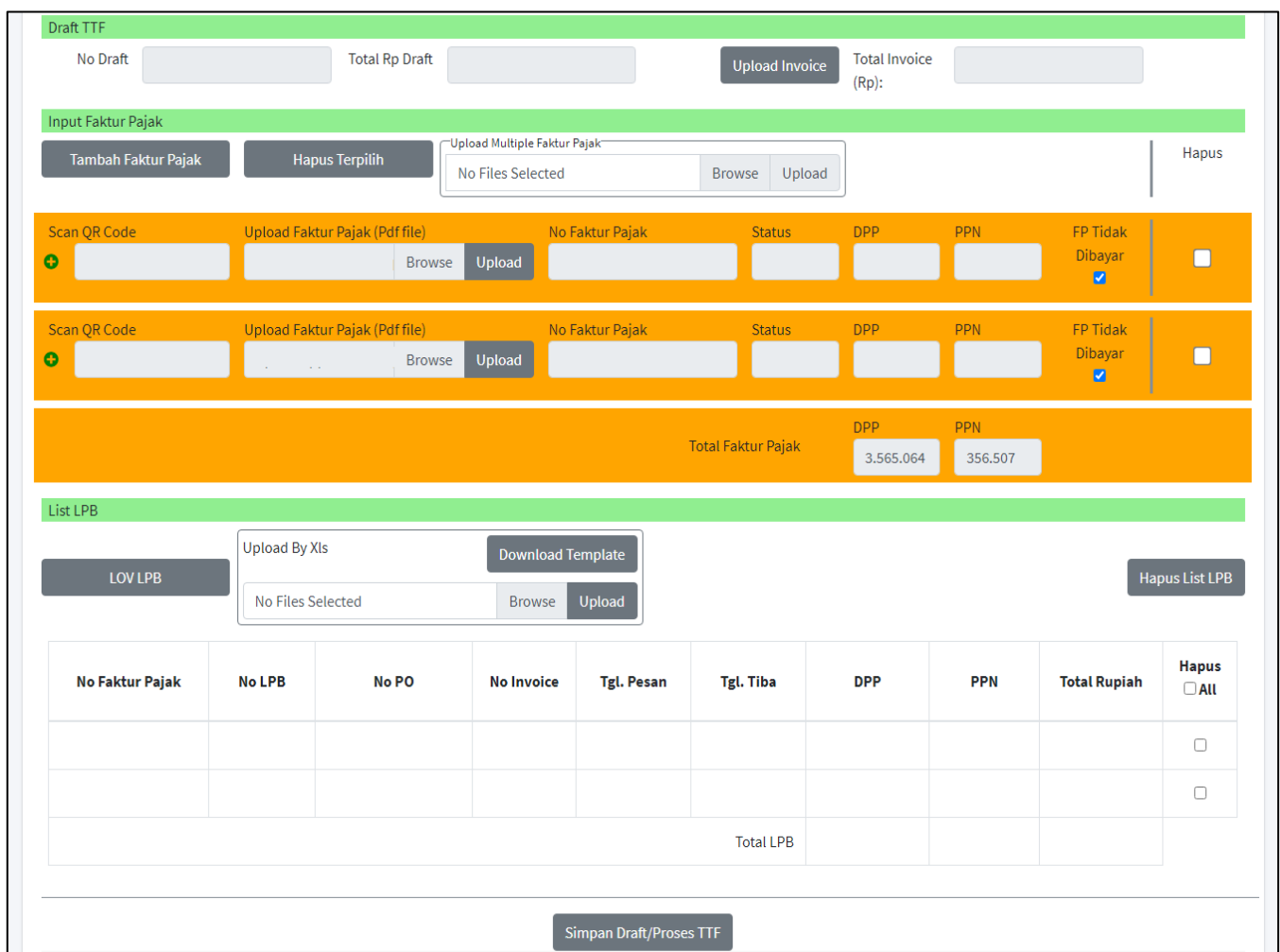


Fig. 3 User interface B2B TTF application

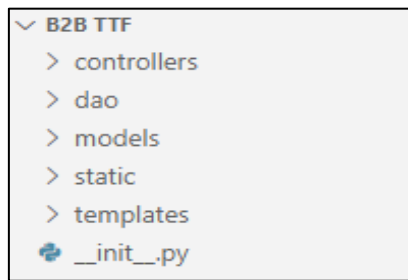


Fig. 4 Flask project structure

For program initialization, Flask has an `__init__.py` file which plays an essential role in defining configurations and accommodating routes for the program. The components of the `__init__.py` file can be seen in Fig. 5. It can be seen in line one to line three that there is the import syntax used to declare the PostgreSQL adapter library, Flask framework, and Flask's Login Manager, which functions to accommodate session data. On line five, we define the project as an object in Flask, for lines six and seven, it works to configure the project and the project declaration to be able to use the session

from the Login Manager, while on line nine, the `dbConnect ()` function is used to configure the PostgreSQL database and in line 13 functions to perform all the initialization functions that are made.

Google Cloud integration on B2B TTF application is based on the advantages of Google Cloud, making data access flexible. One of the services used is the Google App Engine as a platform that provides facilities for hosting Google servers. To host the application, you need the `app.yaml` and `requirements.txt` files on the project. The `app.yaml` component file can be seen in Fig. 6, where it can be seen in line one that a declaration is made for the runtime technology used. The third line defines the instances to be used. F1 is a scaling service that functions to set the minimum and the maximum number of instances to be used, latency and concurrent connections. The `env_variables` on line five define environment variables used in the project locale to be recognized by Google App Engine. While the `requirements.txt` contains the libraries needed by the application and the version used [9].

```

1. import psycopg2
2. from flask import Flask
3. from flask_login import LoginManager
4.
5. app = Flask(__name__)
6. app.config.from_pyfile('settings.py')
7. login_manager = LoginManager(app)
8.
9. def dbConnect():
10.     connection =
        psycopg2.connect(user=app.config["USER"],password=app.config["PA
        SS"],host=app.config["HOST"],port=app.config["PORT"],database=ap
        p.config["DATABASE"])
11.     return connection
12.
13. from b2b.controllers import (routes)
    
```

Fig. 5 Init.py file components

```

1. runtime: python38
2.
3. instance_class: F1
4.
5. env_variables:
6.     bucket_b2bttf: "bucketb2bttf"
7.     db_host_b2bttf: "hostb2bttf"
8.     db_pass_b2bttf: "secretkey"
9.     db_user_b2bttf: "userb2bttf"
10.    db_id_b2bttf: "b2bttf"
11.    db_port_b2bttf: "8888"
    
```

Fig. 6 App.yaml file components

Google Cloud services such as Google Compute Engine are used in the B2B TTF application to perform API requests to validate and get detailed invoice data uploaded by users. The function for making requests to the Google Compute Engine can be seen in Fig. 7. In order to run the requests method function, a library requests is called on line one. The call to the `JSON` function on line two is used to get the return data in the form of `JSON`. Utilizing the `requests` library, on line five a request is made with the `POST` method to the Google Compute Engine by carrying a header parameter in the form of a key to access the API and the body in the form of an invoice file uploaded by the user in `JSON` form.

```

1. import requests
2. import json
3.
4. dataAPI = requests.post(API_LINK, headers=KEY_HEADER, files=files)
5. json.loads(dataAPI.content)

```

Fig. 7 Requests function components to GCE

Application data storage is carried out on Google Cloud Storage so that data can be accessed quickly and maintain stored data security. Bucket is a resource that Google Cloud Storage owns, bucket consists of several objects that can be accessed by several methods [10]. Functions for uploading and downloading data from Google Cloud Storage can be seen in Fig. 8 and Fig. 9. On line one, Fig. 8 and Fig. 9 function to declare library storage from Google Cloud. Line three in Fig. 8 and Fig. 9 function to create objects from library storage, while lines four and five in Fig. 8 and Fig. 9 are used to configure buckets and the location of data in the bucket. Line six functions to upload data which can be seen in Fig. 8, while data download can be seen on line six in Fig. 9.

Google Cloud also provides services for performing OCR. OCR is used to scan the total value on the invoice. One of Google's technologies to perform OCR is Google Vision, which can accept text, the shape of an object, and select a character on an object. Functions for handling characters in data from buckets using Google Vision can be seen in Fig. 10 [11]. It functions to declare the JSON, regex, Google Vision, and Google Storage libraries on lines one through four. Line six defines the file type to be read. Line seven represents the page for grouping JSON output. Line eight serves to create objects in the functions from the Google Vision library. Lines 10 to 18 are used for the layout of input and output files on Google Cloud Storage. Lines 16 to 24 to make OCR service requests with the configuration that has been made. Lines 26 and 27 serve to provide a time lag for the OCR process to complete before uploading the output file to the bucket.

After the B2B TTF application development is complete, application testing is carried out using alpha testing and beta testing. Alpha testing is the initial testing

stage carried out for product validation from all perspectives. Alpha testing helps ensure product readiness before beta testing. Meanwhile, beta testing is carried out to get feedback from users and ensure the product is ready for release [12-13].

Based on the results of alpha testing in Table I, it can be concluded that each function in the B2B TTF application can run according to user requests. The beta testing results, which can be seen in Table II, were carried out by distributing a questionnaire containing questions addressed to users aimed at obtaining feedback from users. Some users will be given access to try the application, which the user will then ask to provide feedback regarding the application that has been tested. The number of respondents involved in filling out the questionnaire was 20 people, consisting of 15 people from the company's IT division and five people from the company's finance division. The index calculation of the questionnaire answers uses a Likert scale.

```

1. from google.cloud import storage
2.
3. client = storage.Client()
4. bucket = client.get_bucket(BUCKETNAME)
5. blob = bucket.blob("myBucket/myFile.pdf")
6. blob.upload_from_filename(FILENAME)

```

Fig. 8 Function to upload to GCS

```

1. from google.cloud import storage
2.
3. client = storage.Client()
4. bucket = client.get_bucket(BUCKETNAME)
5. blob = bucket.blob("myBucket/myFile.pdf")
6. blob.download_to_filename(FILENAME)

```

Fig. 9 Function to download from GCS

```

1. import json
2. import re
3. from google.cloud import vision
4. from google.cloud import storage
5.
6. mime_type = 'application/pdf'
7. batch_size = 2
8. client = vision.ImageAnnotatorClient()
9.
10. feature = vision.Feature(
11.     type=vision.Feature.Type.DOCUMENT_TEXT_DETECTION)
12. gcs_source = vision.GcsSource(uri=gcs_source_uri)
13. input_config = vision.InputConfig(
14.     gcs_source=gcs_source, mime_type=mime_type)
15.
16. gcs_destination = vision.GcsDestination(uri=gcs_destination_uri)
17. output_config = vision.OutputConfig(
18.     gcs_destination=gcs_destination, batch_size=batch_size)
19.
20. async_request = vision.AsyncAnnotateFileRequest(
21.     features=[feature], input_config=input_config,
22.     output_config=output_config)
23. operation = client.async_batch_annotate_files(
24.     requests=[async_request])
25.
26. statusOperation = operation.done()
27. operation.result(timeout=420)
    
```

Fig. 10 OCR function with Google Vision

TABLE I
ALPHA TESTING RESULT

Function Tested	Condition	Result	Test Status
Input TTF	User can do create and update draft.	Success to create and update draft.	Valid
List Draft	User can view the draft and can make edit and delete draft.	Success to show list draft and redirect to edit draft, dan delete draft.	Valid
Upload Invoice	User can do create, update, delete invoice.	Success to create, update, delete invoice.	Valid

TABLE II
QUESTIONNAIRE RESULT

No	Question	STS	TS	N	S	SS
1.	Can the b2b ttf application run as it should?	0	0	3	8	9
2.	Does the b2b ttf application make the invoice exchange process easier?	0	0	9	6	5
3.	Does the list draft menu provide draft information appropriately?	0	0	4	10	6
4.	Is the ttf input process on the b2b ttf website easy to understand and can it run as it should?	0	0	6	10	4
5.	Are the results of the upload invoice is correct?	0	2	4	9	5

TABLE III
QUESTIONNAIRE ASSESSMENT INDEX

Answer	Value
Totally disagree (STS)	1
Disagree (TS)	2
Neutral (N)	3
Agree (S)	4
Totally agreed (SS)	5

TABLE III shows the value according to the questionnaire's answer, which consists of five choices, and each option has a value [14]. With the above questionnaire data, to determine the index as a yardstick for concluding, the calculation is carried out using the formula:

$$Index = \frac{Total\ value}{Maximum\ score} \times 100\% \quad (1)$$

Based on (1), it can be concluded that the index obtained can be categorized into the interval value in TABLE IV [15].

The questionnaire's result is that question one has an index of 86%, so it is included in the strongly agree category. It can be concluded that this system can run according to user demand. The questionnaire's result is that question two has an index of 76%, so that it is included as the agree on category. It can be concluded that this system can simplify the invoice exchange process. The questionnaire's result is that question three have an index of 82%, so it is included in the strongly agree category. It can be concluded that the list draft menu provides appropriate draft information. The questionnaire's result that question four has an index of 78% is included as the agree on category. It can be concluded that users can easily understand the TTF input process in this application. The questionnaire's result is that question five has an index of 77%, so it is included as the agree on category. It can be concluded that the OCR scan results for invoice uploads have a high enough percentage to display the total bill accordingly.

TABLE IV
ASSESSMENT INTERVAL

Value Interval	Category
Index 0% - 19,99%	Totally disagree
Index 20% - 39,99%	Disagree
Index 40% - 59,99%	Neutral
Index 60% - 79,99%	Agree
Index 80% - 100%	Totally agreed

IV. CONCLUSION

In this study, a B2B TTF application was developed using the Flask framework and integrated with Google Cloud. The B2B TTF application has two users, namely employees, to facilitate company accounting and suppliers' management to make it easier to exchange invoices. Based on the test results, it can be concluded that the B2B TTF application can make it easier for users to exchange invoices, and the application of OCR using the Google Vision service can display results that are pretty high with a percentage of 77%. The B2B TTF application can accommodate the TTF input process electronically and help the company's accounting management. Suggestions for further research, the OCR requires image processing to improve the quality of OCR in reading text on images, to increase the percentage to display relevant result.

REFERENCES

- [1] V. R. Vyshnavi and A. Malik, "Efficient Way Of Web Development Using Python and Flask," *Int. J. Recent Res. Asp.*, vol. 6, no. 2, pp. 16-19, 2019.
- [2] G. Prehandayana, W. Yahya, and H. Nurwasito, "Implementasi Struktur Data Dictionary Untuk Sistem Monitoring Perangkat Internet of Things," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3466-3473, 2018.
- [3] E. Riana, "Implementasi Cloud Computing Technology dan Dampaknya Terhadap Kelangsungan Bisnis Perusahaan Dengan Menggunakan Metode Agile dan Studi Literatur," *JURIKOM (Jurnal Ris. Komputer)*, vol.7, no. 3, pp. 439-449, 2020.
- [4] M. Singh, A. Verma, A. Parasher, N. Chauhan, and G. Budhiraja, "Implementation of Database Using Python Flask Framework," *Int. J. Eng. Comput. Sci.*, vol. 8, no. 12, pp. 24890-24893, 2019.
- [5] A. Halim, "Pengembangan Aplikasi Project Time," *JSMSTMik Mikroskil*, vol. 14, no. 1, pp. 69-78, 2013.
- [6] I. N. Abidah, M. A. Hamdani, and Y. Amrozi, "Implementasi Sistem Basis Data Cloud Computing pada Sektor Pendidikan," *KELUWIH J. Sains dan Teknol.*, vol.1, no. 2, pp. 77-84, 2020.
- [7] J. L. Lamattina, "The Impact of Mergers on Pharmaceutical R&D," *Nat. Rev. Drug Discov.*, vol. 10, no. 8, pp. 559-560, 2011.
- [8] M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah, and A. Rokhim, "Design an MVC Model using Python for Flask Framework Development," *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, no. Mvc, pp. 214-219, 2019.
- [9] Google Cloud Support, "Quickstart for Python 3 in the

- App Engine Standard Environment,” 2021. [Online]. Available: <https://cloud.google.com/appengine/docs/standard/python3/quickstart>. [Accessed: 15-Apr-2021].
- [10] Google Cloud Support, “Buckets,” 2021. [Online]. Available: https://cloud.google.com/storage/docs/json_api/v1/buckets. [Accessed: 15-Apr-2021].
- [11] Google Cloud Support, “Annotate a batch of files in Cloud Storage,” 2021. [Online]. Available: https://cloud.google.com/vision/docs/samples/vision-batch-annotate-filesgcs#vision_batch_annotate_files_gcs-python. [Accessed: 15-Apr-2021].
- [12] J. Abadi, B. D. D. Arianti, and R. H. Wirasasmita, “Pengembangan Media Lembar Kerja Siswa (LKS) Berbasis Web Pada Mata Pelajaran Jaringan Dasar,” *EDUMATIC J. Pendidik. Inform.*, vol. 2, no. 1, pp. 42-51, 2018.
- [13] S. Masripah and L. Ramayanti, “Penerapan Pengujian Alpha dan Beta Pada Aplikasi,” *J. SWABUMI*, vol. 8, no. 1, pp. 100–105, 2020.
- [14] B. Sugiantoro, “Pengembangan Qr Code Scanner Berbasis Android Untuk Sistem Informasi Museum Sonobudoyo Yogyakarta,” *Telematika*, vol. 12, no. 2, pp.134–145, 2015.
- [15] D. Taluke, R. S. M. Lakat, and A. Sembel, “Analisis Preferensi Masyarakat Dalam Pengelolaan Ekosistem Mangrove Di Pesisir Pantai Kecamatan Loloda Kabupaten Halmahera Barat,” *Spasial*, vol. 6, no. 2, pp. 531–540, 2019.

