

IMPLEMENTASI GOOGLE DRIVE CLOUD STORAGE PADA SISTEM REPOSITORI AL-DARING

Ketut Agus Seputra¹, A.A. Gede Yudhi Paramartha², I Nyoman Saputra Wahyu Wijaya³

^{1, 2, 3}Jurusan Teknik Informatika, Fakultas Teknik dan Kejuruan, Universitas Pendidikan Ganesha
Jalan Udayana, Singaraja, Indonesia

e-mail: agus.seputra@undiksha.ac.id¹, yudhi.paramartha@undiksha.ac.id²,
wahyu.wijaya@undiksha.ac.id³

Received : November, 2022

Accepted : April, 2022

Published : April, 2022

Abstract

One of the steps of accreditation for institutions and study programs during the epidemic is online assessment (AL-Daring). The availability of suitable information and communication technologies to enable the AL-Daring process is one of the consequences of AL-Daring. Supporting data must be provided in the form of a soft copy in the University Information System that assessors may access appropriately. How the Information System can help Study Program Managers and Accreditation Applicants complete documentation more quickly and efficiently is also crucial. It's also important to note that each Study Program and Study Program Manager will upload a large number of papers of various types and sizes. As a result, various factors that affect the quality of information systems, such as data and storage space availability, ease of access, and timeliness, must be satisfied. The AL-Daring Information System is being developed using agile methodologies in the aim of completing it faster so that it can be utilized right away and adapt to user needs. To perform successfully, this system was built using the Codeigniter framework and Google Drive Cloud Storage as a document storage medium via the Google Drive API service. SUS rates the repository system an excellent score on an adjective rating claiming the system is very useful and well-received by users.

Keywords: Information Systems, Google Drive Cloud Storage, Google Drive API

Abstrak

Assesmen Lapangan secara Daring (AL-Daring) merupakan salah satu tahapan yang dilalui oleh Universitas maupun Program Studi yang melakukan Akreditasi dimasa pandemi. Salah satu konsekuensi dari AL-Daring adalah tersedianya teknologi informasi dan komunikasi yang memadai untuk memfasilitasi proses AL-Daring. Data dukung sesuai dokumen usulan akreditasi harus tersedia dalam bentuk soft copy dalam Sistem Informasi Perguruan Tinggi yang dapat diakses secara baik oleh asesor. Tidak kalah penting juga adalah bagaimana Sistem Informasi tersebut mampu menghadirkan kemudahan bagi Pengelola Prodi maupun pengusul Akreditasi untuk melengkapi dokumen secara lebih cepat dan efisien. Penting juga dipertimbangkan akan ada banyak dokumen dengan berbagai tipe dan ukuran yang akan diunggah oleh masing-masing Prodi maupun Pengelola Prodi. Beberapa hal yang mempengaruhi kualitas Sistem Informasi seperti ketersediaan data dan ruang penyimpanan, kemudahan akses, serta ketepatan waktu wajib dipenuhi. Metode Agile digunakan untuk mengembangkan Sistem Informasi AL-Daring dengan harapan sistem dapat selesai lebih cepat agar segera dapat digunakan serta menyesuaikan dengan kebutuhan pengguna. Sistem Informasi Repositori Dokumen AL-Daring yang dikembangkan menggunakan framework Codeigniter dengan memanfaatkan Google Drive Cloud Storage sebagai media penyimpanan dokumen melalui layanan Google Drive API dapat berfungsi dengan baik. Pengujian dengan SUS memberikan penilaian Exellent pada adjective

rating yang menyatakan bahwa sistem repositori sangat bermanfaat dan diterima dengan baik oleh pengguna.

Kata Kunci: Sistem Informasi, Google Drive, Cloud Storage, Google Drive API, SUS

1. PENDAHULUAN

Perguruan Tinggi sebagai sebuah lembaga pendidikan di Indonesia dalam pengelolaannya wajib mengikuti Akreditasi sesuai kaidah dan ketentuan yang sudah diatur oleh Pemerintah. Akreditasi sebagai sebuah kegiatan penilaian untuk menentukan kelayakan Perguruan Tinggi maupun Program Studi berdasarkan kriteria yang mengacu pada Standar Nasional Pendidikan Tinggi. Disamping itu, akreditasi juga menjadi salah satu proses penjaminan mutu secara eksternal baik pada bidang akademik maupun non-akademik, yang diharapkan dapat melindungi kepentingan mahasiswa maupun masyarakat secara luas [1][2]. Sebagaimana telah diatur dalam peraturan BAN-PT, bahwa proses akreditasi harus dilandasi oleh prinsip independen, akurat, obyektif, transparan, dan akuntabel. Oleh karena itu asesor lapangan memiliki independensi untuk melakukan penilaian secara akurat dan obyektif sesuai dengan kenyataan dan didasari bukti sah yang ditemukan di lapangan. Proses penilaian dilakukan melalui Assesmen Lapangan yang dilakukan oleh asesor dengan mengunjungi Perguruan Tinggi sesuai dengan jadwal dan agenda kegiatan. Namun terhitung sejak maret 2019 kunjungan ke Perguruan Tinggi tidak dapat dilakukan karena adanya pandemi Covid-19. Hal ini tentu berdampak terhadap Perguruan Tinggi yang sangat menantikan proses dan hasil Assesmen Lapangan sebagai salah satu bentuk pertanggung jawaban dan pengakuan terkait capaian kinerja yang telah diperoleh. Berkaitan dengan hal tersebut maka BAN-PT memutuskan untuk melaksanakan Assesmen Lapangan secara Daring (AL-Daring).

Salah satu konsekuensi dari AL-Daring adalah tersedianya teknologi informasi dan komunikasi yang memadai untuk memfasilitasi proses AL-Daring. Data dukung sesuai dokumen usulan akreditasi harus tersedia dalam bentuk *soft copy* dalam Sistem Informasi Perguruan Tinggi yang dapat diakses secara baik oleh asesor [3]. Disamping untuk menjaga validitas data dan informasi, Sistem Informasi tersebut tentu harus mampu membantu asesor untuk memvalidasi bukti dengan mudah, sehingga proses AL-Daring dapat berjalan dengan lebih

efektif. Tidak kalah penting juga adalah bagaimana Sistem Informasi tersebut mampu menghadirkan kemudahan bagi Pengelola Prodi maupun pengusul Akreditasi untuk melengkapi dokumen secara lebih cepat dan efisien [4].

Penataan dokumen akreditasi sesuai standar Borang Akreditasi menjadi fokus utama dalam pengembangan sistem repositori. Dokumen yang telah tertata dengan teratur tentu akan mempermudah dalam proses temu kembali [5]. Disamping itu pula, pelaksanaan akreditasi yang membutuhkan dokumen secara lengkap mulai dari tingkat program studi, fakultas, hingga universitas menuntut sistem repositori harus dikembangkan secara terpusat [6]. Pengarsipan dokumen secara terpusat tentu juga akan mengurangi redundansi dari dokumen yang diunggah. Penting juga dipertimbangkan akan ada banyak dokumen dengan berbagai tipe dan ukuran yang akan diunggah oleh masing-masing Prodi maupun Pengelola Prodi. Sehingga beberapa hal yang mempengaruhi kualitas Sistem Informasi seperti ketersediaan data dan ruang penyimpanan, kemudahan akses, serta ketepatan waktu wajib dipenuhi. Pengembangan media penyimpanan dengan konsep *server clustering* membutuhkan *load balancing* yang dapat menimbulkan tantangan baru pada distribusi dokumen pada server dan peningkatan pada biaya operasional. Oleh karena itu, layanan *cloud computing* dapat menjadi salah satu alternatif untuk mengatasi masalah ini. Melalui layanan *cloud storage* memungkinkan sebuah aplikasi melakukan penyimpanan file pada *cloud storage* atau melakukan sinkronisasi server lokal dengan *cloud storage*. Namun dengan bertambahnya file yang diunggah akan membutuhkan media penyimpanan dengan kapasitas yang lebih besar. Hal ini tentu akan menyebabkan peningkatan terhadap biaya sewa layanan *cloud storage* [7].

Berdasarkan atas beberapa uraian masalah tersebut, maka Sistem Informasi Repositori Dokumen AL-Daring dikembangkan dengan memanfaatkan *Google Drive* sebagai media penyimpanan dokumen melalui layanan *Google Drive API*. Kapasitas penyimpanan gratis yang disediakan oleh *google drive* sebesar 15 GB dan didukung oleh keamanan serta kemudahan

interoperabilitas google drive tentu akan meningkatkan efisiensi dan efektivitas sistem repositori.

2. METODE PENELITIAN

2. 1. Tinjauan Pustaka

Cloud Storage merupakan sebuah trend pada komputasi berbasis internet yang melibatkan banyak komputer untuk bekerja bersama demi tujuan tertentu. Disamping itu *cloud storage* memungkinkan user, pelanggan, atau perusahaan menyimpan data tanpa menggunakan sistem penyimpanan lokal. Data dan informasi yang telah tersimpan dapat diakses secara sempurna melalui *client services* dan *network connectivity*. Terdapat tiga jenis model dalam layanan cloud storage yakni [8].

- Software as a Services (SaaS)* menyediakan aplikasi penyimpanan yang siap digunakan oleh pengguna. Seperti misalnya Google Drive.
- Platform as a Services (PaaS)* dimana model ini menyediakan aplikasi termasuk lingkungan pengembangannya yang bisa disesuaikan dengan kebutuhan pengguna.
- Infrastructure as a Services (IaaS)* menyediakan media penyimpanan dan sumber daya komputasi yang dapat memberikan solusi bisnis yang dapat disesuaikan.

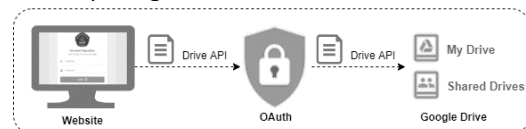
Cloud storage menghadirkan fitur kepastian akan ketersediaan (*availability*), data aman dari kegagalan infrastruktur (*durability*), kemudahan dan kecepatan akses (*performance*) [9]. Disamping fitur tersebut, dilihat dari sudut pandang kapasitas dan kemudahan akses terdapat beberapa keuntungan dari penyimpanan pada *cloud storage* yakni.

- Sederhana dalam penggunaan yang didukung oleh lingkungan pengembangan dan infrastruktur yang selalu mengalami peningkatan.
- Penyimpanan *cloud* berguna untuk mengurangi biaya kepemilikan terutama dalam model SaSS [10].
- Menyimpan di *cloud* jauh lebih baik daripada penyimpanan konvensional lainnya. Sebagai contoh pengguna tidak perlu menyediakan media penyimpanan fisik dengan kapasitas besar termasuk backup data.
- Peningkatan ataupun perbaikan terkait teknologi dan infrastruktur tidak

mempengaruhi layanan yang diberikan kepada pengguna.

- Pengguna dapat memilih layanan *cloud* atau kombinasi menggunakan *cloud* dengan media penyimpanan lokal.

Google drive adalah salah satu aplikasi penyimpanan berbasis *cloud* yang disediakan oleh google. Kemampuan *google drive* dalam integrasi data dengan aplikasi google lainnya termasuk juga aplikasi pihak ketiga membuat pengguna google drive meningkat. Hal ini juga didukung kapasitas penyimpanan secara gratis sebesar 15GB dan tidak terbatas untuk pengguna *Gsuite* [11]. Google drive memungkinkan berbagi dokumen dan data melalui fitur Google Drive API. Fitur Rest API yang dimiliki memungkinkan pengembangan aplikasi dapat memanfaatkan penyimpanan google drive untuk menyimpan file statis seperti terlihat pada gambar 1.



Gambar 1. Google Drive API Relationship [12]

Software Development Kit (SDK) Google Drive API termasuk didalamnya HTTP API dapat digunakan oleh pengembang aplikasi untuk melakukan penyimpanan, integrasi, dan akses file dari aplikasi yang sedang dibangun [12]. Disamping itu, pengguna atau pengembang aplikasi dapat menggunakan beberapa media penyimpanan *cloud* yang dapat diintegrasikan dengan aplikasi [13].

Pengujian aplikasi yang dikembangkan penting dilakukan untuk mengukur performa aplikasi, tingkat kegunaannya, serta tingkat keterimaannya oleh pengguna. SUS adalah suatu tools yang paling umum digunakan untuk mengukur tingkat kegunaan dari suatu produk[14]. Perhitungan skor yang sederhana, tidak membutuhkan biaya dalam penggunaannya, dengan jumlah sampel yang yang dibutuhkan tidak terlalu besar membuat metode evaluasi ini mudah dimengerti. SUS mengolah data yang didapat dari skenario pengujian untuk mengetahui presentase tingkat keberhasilan dari tiap skenario oleh masing-masing pengguna dalam skala 0-100. Adapun perhitungan skor kuisioner SUS adalah sebagai terdapat pada persamaan (1)[15].

$$SUS = \frac{((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10)) * 2.5}{10}$$

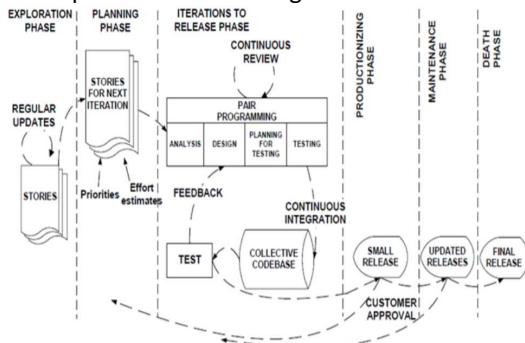
(1)

2. 2. Data

Data yang digunakan dalam penelitian adalah data kualitatif dan kuantitatif. Data kualitatif diperoleh melalui tahap observasi dan literatur review yang terdiri dari masalah dan tujuan yang mendasari pengembangan sistem, dokumentasi pengimplementasian Google Drive API, serta profil pengguna sistem. Data Kuantitatif diperoleh dari hasil pengujian sistem berupa hasil kuisioner SUS dan *Black Box Testing*.

2. 3. Metode Pengembangan Sistem

Pemilihan metode pengembangan perangkat lunak menjadi suatu hal yang penting dan menentukan bagi keberhasilan pengembangan. Metode Agile banyak digunakan dalam industri untuk menghasilkan perangkat lunak baru dengan lebih cepat [16]. Pemilihan metode agile dapat membantu pengembangan fokus pada fokus pada proses bisnis, sehingga resiko pengembangan sistem dapat dikurangi dan mengoptimalkan value bisnis [4]. Disamping itu, metode agile tanggap dengan perubahan yang membuat tim lebih responsif terhadap proses bisnis yang dapat berubah, sehingga kualitas perangkat lunak menjadi lebih baik sesuai dengan kebutuhan bisnis [17]. Penggunaan langsung terhadap sistem yang sedang dikembangkan juga menjadi salah satu dasar dalam pemilihan metode Agile.



Gambar 2. Siklus *Extreme Programming* [18]

Extreme Programming (XP) merupakan salah satu metode Agile yang banyak digunakan untuk pengembangan perangkat lunak yang melibatkan tim kecil hingga medium. Metode ini menekankan pada keterbukaan terhadap kebutuhan bisnis dan kerjasama tim yang kuat [18]. Adapun siklus pengembangan perangkat lunak dengan metode XP sesuai gambar 2 yakni.

- Exploration Phase* merupakan tahapan untuk menggali kebutuhan pengembangan perangkat lunak. Pada tahapan ini kebutuhan pengguna, arsitektur sistem,

prangkat pendukung, termasuk infrastruktur teknologi telah selesai didefinisikan. Termasuk memahami alur dari sebuah proses AL-Daring.

- Planning Phase* fokus pada penetapan fungsional sistem, jadwal pengembangan, pembagian tugas dalam tim, termasuk *deadline* pembuatan dalam setiap tahapan.
- Iterations to Release Phase* merupakan tahapan inti dari pengembangan perangkat lunak. Pada tahap ini terdiri dari beberapa proses dasar dalam pengembangan perangkat lunak yakni mulai dari analisis, desain, coding, hingga pengujian. Tahapan ini yang selalu akan mengalami pengulangan kegiatan jika terdapat perubahan atau penambahan proses bisnis.
- Productionizing Phase* merupakan tahapan *deployment* perangkat lunak dalam versi beta. Pada versi beta, perangkat lunak telah memenuhi *Minimum Product Variable* sesuai kebutuhan bisnis.
- Maintenance Phase* merupakan sebuah siklus alami dari sebuah pengembangan perangkat lunak. Dalam fase ini fungsionalitas baru dibangun saat sistem sedang digunakan. Oleh karena itu penambahan fitur harus dilakukan secara berhati-hati.
- Death Phase* adalah tahap akhir dari siklus pengembangan perangkat lunak. Terdapat dua situasi yang memungkinkan pengembangan perangkat lunak telah mencapai tahap ini. Pertama, pelanggan merasa puas atas perangkat lunak yang dihasilkan telah memenuhi seluruh kebutuhan bisnis. Kedua, pelanggan membutuhkan fitur yang tidak dapat dikembangkan secara ekonomis. Dalam situasi seperti ini, sebaiknya menghentikan pengembangan perangkat lunak yang biasa disebut sebagai *entropic death system*.

2. 4. Perancangan Sistem

2.4.1 Analisis

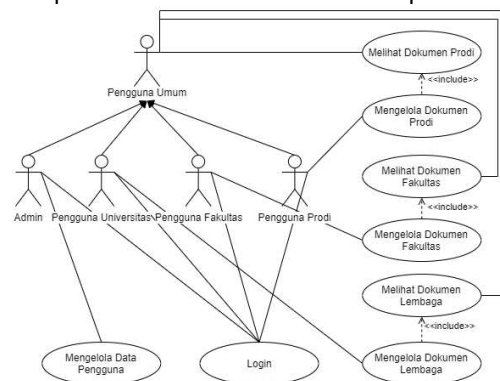
Tahap analisis diawali oleh eksplorasi data baik itu melalui wawancara, observasi proses AL-Daring, maupun studi literatur terkait tatakelola dokumen akreditasi. Tujuan dari tahapan eksplorasi untuk mengidentifikasi permasalahan yang terjadi, kebutuhan pengguna, tujuan pengembangan aplikasi, sasaran, sumber daya, termasuk pengguna aplikasi [19]. Dari hasil eksplorasi, teridentifikasi

- a. Tersedia teknologi informasi dan komunikasi yang memadai untuk memfasilitasi terlaksana AL-Daring.
- b. Data dukung sesuai dokumen usulan akreditasi harus tersedia dalam bentuk *soft copy* dalam Sistem Informasi Perguruan Tinggi yang dapat diakses secara baik oleh asesor.
- c. Data dukung terintegrasi mulai dari tingkat universitas, fakultas, dan program studi. Sehingga dapat mengurangi redudansi dari dokumen yang diunggah.
- d. Terdapat beberapa level pengguna yaitu prodi, fakultas, universitas, dan asessor.
- e. Kapasitas penyimpanan yang dapat menyimpan seluruh dokumen.
- f. Sistem memiliki tingkat *availability* dan *performance* yang tinggi terkait akses dan ketersediaan dokumen.

The diagram illustrates the system architecture. On the left, a box labeled 'Sistem Informasi: Sistem Repositori Dokumen' contains icons for 'Codeigniter' and 'Mysql'. Below this box is a screenshot of a web application interface. An arrow labeled 'Drive API' points from the system to a central shield icon labeled 'OAuth'. Another arrow labeled 'Drive API' points from the 'OAuth' icon to a box on the right labeled 'Media Penyimpanan File Statis'. This box contains icons for 'Drive Universitas', 'Drive Fakultas 1', 'Drive Fakultas Dist.', 'Drive Prodi 1', and 'Drive Prodi Dist.'.

Bertitik tolak dari analisis kebutuhan pengguna, maka dibuat sebuah sistem repositori yang dikembangkan berbasis website dengan MySQL sebagai *Database Management System*.

- Terdapat tiga level pengguna assesor, prodi, fakultas, dan lembaga.
- Pengguna dapat mengunggah dokumen data dukung dalam berbagai tipe dan ukuran file sesuai dengan level pengguna.
- Sistem menggunakan media penyimpan beberapa google drive yang dimiliki oleh prodi, fakultas dan lembaga yang terlibat dalam AL-Daring.
- Terkait kerahasiaan data, prodi dan fakultas hanya bisa melakukan penambahan dokumen pada sistem yang terhubung dengan google drive sesuai dengan credentials Google Drive API masing-masing lembaga baik prodi dan fakultas.
- Seluruh pengguna termasuk assesor dapat melihat dokumen AL-Daring seluruh prodi pada halaman beranda sistem repositori.

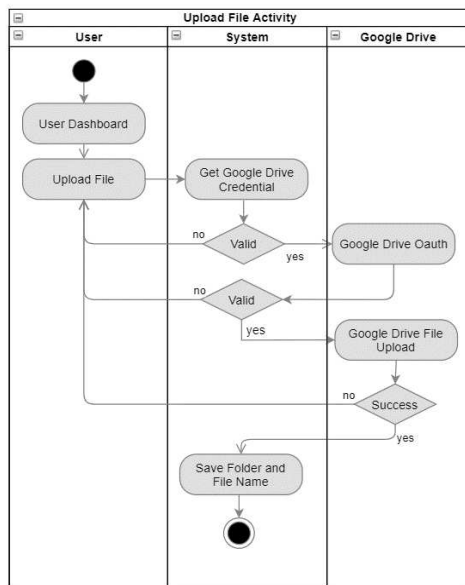


Kebutuhan fungsional tersebut selanjutnya dimodelkan menggunakan *Use Case Diagram* guna mengetahui akses masing-masing aktor terhadap sistem. Adapun kewenangan akses aktor dapat dilihat pada gambar 4.

Perancangan dilakukan secara detail menyesuaikan dengan kebutuhan pengguna dan arsitektur sistem yang telah didefinisikan pada tahap analisis. Beberapa dokumen terkait bagan pengembangan sistem sangat diperlukan guna memperjelas fitur dan alur kerja sistem. Setiap siklus XP selalu melalui tahapan perancangan ketika terdapat penambahan atau perubahan proses bisnis. Sehingga pada setiap siklus, selalu menghasilkan sebuah dokumen yang detail terkait perancangan sistem yang dikembangkan pada siklus tersebut.

- SINTECH Journal | 53

Gambar 5 merupakan diagram aktivitas yang dapat menceritakan alur kerja dari sebuah proses.



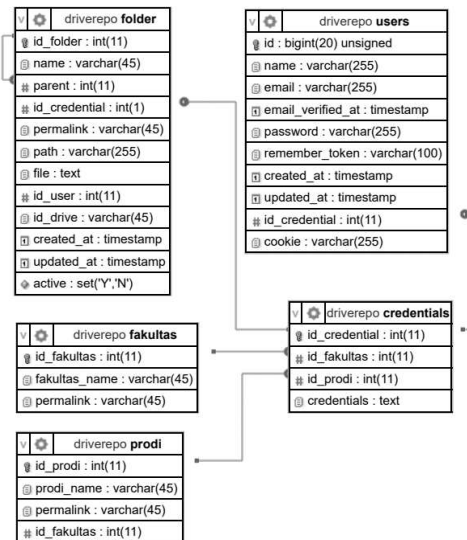
Gambar 5. Diagram Aktivitas Unggah Berkas

Berikut dipaparkan mengenai aktivitas unggah berkas pada sistem. Unggah berkas dapat dilakukan oleh masing-masing pengguna dari setiap lembaga yang terdaftar. Untuk dapat melakukan pengunggahan berkas, pengguna wajib login untuk memperoleh akses dan *Credential Google Drive Api*. Pengguna dapat mengunggah beberapa berkas dengan berbagai tipe file yang tidak dibatasi melalui form upload berkas. Proses unggah berkas dimulai dengan memvalidasi *credential login* dari pengguna. *Credential* tersebut digunakan untuk memperoleh autentikasi dari *Google Oauth*. Setelah dinyatakan valid dan terdaftar pada *Google Drive API*, maka berkas baru dapat diunggah pada masing-masing *Google Drive* sesuai dengan *credential* masing-masing pengguna. Proses unggah berkas dapat dinyatakan selesai setelah memperoleh respon sukses dari *Google Drive API*, respon tersebut seperti folder id, folder name, dan file name akan disimpan pada *MySQL* sesuai fakultas dan prodi yang tersimpan pada *credential*. Penyimpanan respon kedalam database lokal ditujukan untuk mempermudah proses temu kembali file pada sistem repositori.

b. Entity Relationship Diagram (ERD)

ERD merupakan sebuah diagram struktural yang dapat menggambarkan secara detail *entity* atau komponen data dan hubungannya

melalui *relationship set*, serta jumlah anggota dalam dalam suatu hubungan melalui *constraints* [20]. Adapun detail hubungan setiap entitas dalam sistem repositori digambarkan dalam *Physical Data Model* seperti pada gambar 6.



Gambar 6. Physical Data Model

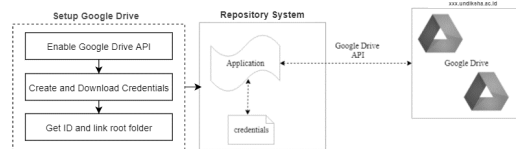
JSON Credentials tersebut didapat dari proses aktivasi *Google Drive API* seperti pada gambar 7. Selain menyimpan data *credentials Google Drive*, entitas ini juga menyimpan hubungan antara users dengan fakultas dan prodi. Dari hubungan inilah dapat diperoleh sebuah konfigurasi *credentials Google Drive API* yang digunakan pada proses unggah berkas. Termasuk juga nanti hubungan kelima entitas ini yang digunakan dalam menampilkan berkas.

3. HASIL DAN PEMBAHASAN

3. 1. Implementasi

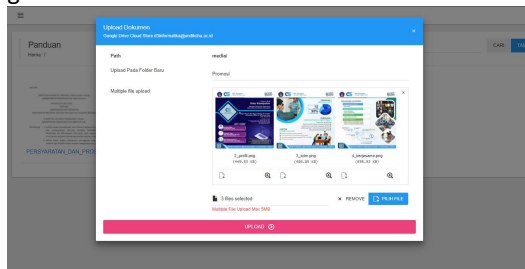
Sistem repositori dikembangkan menggunakan framework PHP yakni Codeigniter dengan MySQL sebagai *Database Mangement System*. Sistem dikembangkan sesuai dengan arsitektur dan rancangan sistem. Langkah awal dalam pengembangan sistem adalah membuat tampilan situs yang *responsive* terhadap *devices* pengguna, serta interaktif. Oleh karena itu, pada bagian *frontend* sistem menggunakan *framework bootstrap CSS* dan *Jquery*. Melanjutkan ke tahap berikutnya adalah menyiapkan logika otorisasi pengguna sesuai level akses masing-masing. Proses otorisasi pengguna adalah proses penting yang langsung berhubungan dengan autentikasi *Oauth Google Drive* sesuai dengan akun *Google Drive* masing-masing lembaga. Salah satu fungsi basis data

disini adalah menyimpan data pengguna beserta *credentials* Google Drive yang digunakan. *Credentials* diperoleh dengan mengaktifkan fitur *Google Drive API* pada setiap *google drive* yang digunakan. Disamping itu, pengguna juga harus menyiapkan sebuah folder *root* pada *Google Drive* untuk menampung seluruh berkas yang akan diunggah, serta mengaktifkan fitur *share link* dengan mode *viewer* untuk menampilkan seluruh isi dari folder *root* tersebut. Adapun alur penggunaan *credentials* seperti pada gambar 7.



Gambar 7. Konfigurasi Credentials

Langkah berikutnya adalah membuat modul untuk unggah berkas pada sistem. Semua file yang diunggah tersimpan pada media penyimpanan *Google Drive* sesuai dengan *credentials* dan folder *root* yang digunakan. Untuk mempermudah manajemen folder dan menu, seluruh data folder dan nama file, termasuk id folder pada *Google Drive* disimpan dalam Basis Data. Sehingga penting, pada setiap proses manajemen folder dan file terjadi sinkronisasi antara Basis Data dengan folder fisik yang dibuat pada *Google Drive*. Adapun tampilan dari form upload file seperti pada gambar 8.



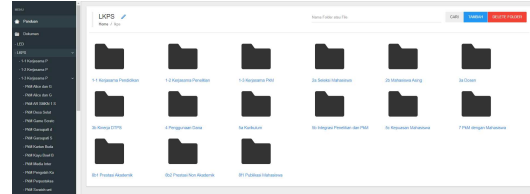
Gambar 8. Form Modal Unggah Berkas

Tahap terakhir dari pengembangan sistem repositori ini adalah proses temu kembali berkas yang telah diunggah. Untuk itu pada halaman detail folder dilengkapi dengan *tree menu* dan beberapa fitur lainnya seerti.

- Fitur pencarian file atau folder oleh admin, assesor dan pengguna umum.
- Fitur tambah folder dan file untuk admin prodi, fakultas maupun lembaga. Fitur ini dapat digunakan untuk unggah banyak file dalam sekali unggah.

- Fitur rename folder untuk melakukan perubahan nama folder oleh admin.
- Fitur delete folder atau file untuk melakukan hapus folder dan isi didalamnya oleh admin.

Adapun tampilan dari halaman yang dimaksud terdapat pada gambar 9.



Gambar 9. Halaman Detail Folder

3. 2. Pengujian

a. Pengujian Performa

Pengujian performa sistem ditujukan untuk menguji kecepatan unggah berkas terhadap sistem. Pengujian dilakukan dengan membandingkan waktu dan ukuran berkas yang diunggah menggunakan *server On-Premise (OP)* dan *Server Google Drive*. Adapun hasil pengujian dapat dilihat pada tabel 1.

Tabel 1: Hasil Pengujian Performa

No	Server	Qty	Size (MB)	Time (s)
1	OP	1	4.56	8.06
2	OP	2	4.56	13.5
3	OP	4	0.5	9.8
4	OP	6	0.5	13
5	Drive	1	4.56	6.4
6	Drive	2	4.56	15
7	Drive	4	0.5	11
8	Drive	6	0.5	12

Pengujian tersebut memberikan hasil bahwa proses unggah berkas pada setiap skenario pengujian telah berhasil dilakukan. Adapun terkait waktu unggah berkas terdapat perbedaan antara penggunaan server *OP* dan *Google Drive*. Namun dari keseluruhan pengujian seperti pada tabel 1, bahwa waktu yang diperlukan untuk upload file *server google* lebih stabil dan relatif lebih cepat. Hal tersebut tentu juga dipengaruhi oleh kecepatan dan *bandwidth* internet.

b. Black Box Testing

Pengujian *black box* menggunakan metode *behaviour testing* ditujukan untuk memastikan fitur utama yakni login, unggah file maupun folder, update nama folder, pencarian file, dan hapus folder maupun file dapat berjalan dengan baik pada beberapa skenario percobaan. Pengujian dilakukan

dengan 10 skenario unggah file dengan beberapa tipe dan ukuran file yang berbeda.

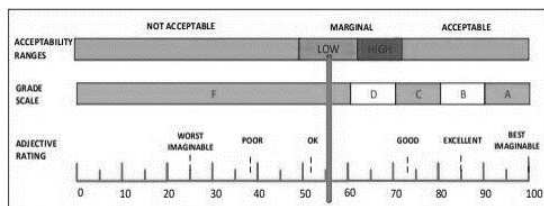
Adapun hasil pengujian dapat dilihat pada tabel 2.

Tabel 2: Hasil Pengujian *Black Box*

Input	Harapan	Simpulan
Login	Halaman pengguna tampil sesuai identitas login lembaga	Berhasil
Unggah file pada folder baru sebanyak 10 kali	File terunggah pada folder yang telah dibuat pada google drive	Berhasil
Unggah file pada folder yang telah tersedia sebanyak 10 kali	File tersimpan pada folder yang telah tersedia	Berhasil
Ubah Nama Folder pada 10 Folder	Nama folder dan permalink dapat dirubah	Berhasil
Pencarian nama file dan folder	File dapat tampil sesuai kriteria pencarian	Berhasil
Delete folder	Folder dan isinya dapat terhapus	Berhasil
Delete File	File terhapus	Berhasil

3. 3. Evaluasi

Responden yang terlibat dalam SUS berjumlah 8 orang tenaga pengajar dari program studi Ilmu Komputer sebagai pengguna sistem. Program studi Ilmu Komputer sebagai salah satu program studi yang menggunakan sistem repositori untuk mendukung proses Assesment Lapangan Daring.



Gambar 10. SUS Performance Chart [21]

Berdasarkan atas kuisioner SUS yang diberikan kepada responden diperoleh rata-rata skor 84.37 dengan nilai tertinggi 97.5 dan nilai terendah 67.5. Oleh karena itu, sesuai *SUS Performance Chart* pada gambar 10, maka skor rata-rata SUS 84.37 memperoleh nilai Exellent untuk tingkat kegunaan, grade B untuk performa aplikasi, dan Acceptable bagi pengguna.

4. KESIMPULAN

Google Drive Cloud Storage melalui layanan Google Drive API V2 dapat digunakan dengan baik sebagai media penyimpanan dokumen pada sistem repositori AL Daring. Hal tersebut dapat dilihat dari pengujian performa yang dilakukan. Hasil pengujian black box juga menyatakan seluruh fitur yang tersedia pada sistem repositori mulai dari login pengguna, upload folder maupun file, ubah nama folder, pencarian, hingga menghapus folder maupun file telah berjalan dengan baik. Penilaian Exellent pada adjective rating SUS telah

menyatakan bahwa sistem repositori sangat bermanfaat dan diterima dengan baik oleh pengguna.

Fitur lain dari google drive yakni google API dapat digunakan dalam penelitian selanjutnya untuk mengembangkan interoperabilitas pertukaran data dan dokumen antar sistem menggunakan middleware google API. Pengembangan sistem terintegrasi ini tentu akan sangat bermanfaat terutama dalam manajemen data dan file untuk keperluan arsip dalam instansi pemerintahan, sekolah, dan badan usaha milik desa.

5. DAFTAR PUSTAKA

- [1] Undang-Undang Republik Indonesia No 12 Tahun 2012, *UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 12 TAHUN 2012 TENTANG PENDIDIKAN TINGGI*. 2012.
- [2] A. Primadewi and M. Hanafi, "Pengelolaan Data Terintegrasi Berdasarkan Instrumen Akreditasi Perguruan Tinggi 3.0 Menggunakan Zachman Framework," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 5–10, 2020.
- [3] B. Akreditasi and N. Perguruan, "Panduan Asesmen Lapangan Secara Daring," no. September, 2020.
- [4] A. Kamran, L. Happy, and T. Palysu, "Pengembangan Sistem Informasi Repository Data Akreditasi Institut Teknologi Kalimantan," vol. 6, no. 2, pp. 200–209, 2020.
- [5] N. R. Radliya and R. Sidik, "Rancang Bangun Sistem Repository Akreditasi Program Studi Manajemen Informatika," *J. Manaj. Inform.*, vol. 8, no. 2, 2018.
- [6] M. R. Haroki, H. Ajie, and M. fick. Duskarnaen, "Pengembangan Sistem Repository Dokumen Akreditasi Institusi

- Perguruan Tinggi di Universitas Negeri Jakarta.pdf." PINTER, Jakarta, 2019.
- [7] I. Sontana, A. Rahmatulloh, and A. N. Rachman, "Application Programming Interface Google Picker Sebagai Penyimpanan Data Sistem Informasi Arsip Berbasis Cloud," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 1, pp. 25–32, 2019.
 - [8] G. Kulkarni, R. Sutar, and J. Gambhir, "Cloud Computing - Storage as Service," *Int. J. Eng. Res. Appl.*, vol. 2, no. 1, pp. 945–950, 2012.
 - [9] P. A. Abdalla, "Advantages to Disadvantages of Cloud Computing for Small-Sized Business," in *International Symposium on Digital Forensics and Security (ISDFS)*, 2019.
 - [10] J. Wu, L. Ping, X. Ge, W. Ya, and J. Fu, "Cloud storage as the infrastructure of Cloud Computing," *Proc. - 2010 Int. Conf. Intell. Comput. Cogn. Informatics, ICICCI 2010*, pp. 380–383, 2010.
 - [11] S. Tata *et al.*, "Quick access: Building a smart experience for google drive," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1296, pp. 1643–1651, 2017.
 - [12] Google Developers, "Introduction to Google Drive API," *Google*. [Online]. Available: <https://developers.google.com/drive/api/v2/about-sdk>. [Accessed: 01-Jun-2021].
 - [13] K. Ibnutama, H. Winata, and M. Hutasuhtut, "Web-Based College Student Assignment File Collection Application Using Google Drive API," vol. 3, no. 2, pp. 34–40, 2019.
 - [14] A. Bangor, P. T. Kortum, and J. T. Miller, "An Empirical Evaluation of the System Usability Scale," *Int. J. Human-Computer Interact.*, vol. 24, no. 6, pp. 574–594, Jul. 2008.
 - [15] B. Beny, H. Yani, and G. M. Ningrum, "Evaluasi Usability Situs Web Kemenkumham Kantor Wilayah Jambi dengan Metode Usability Test dan System Usability Scale," *Res. Comput. Inf. Syst. Technol. Manag.*, vol. 2, no. 1, p. 30, 2019.
 - [16] V. Venkatesh, J. Y. L. Thong, F. K. Y. Chan, H. Hoehle, and K. Spohrer, "How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills," *Inf. Syst. J.*, vol. 30, no. 4, pp. 733–761, 2020.
 - [17] S. Ghobadi and L. Mathiassen, "Perceived barriers to effective knowledge sharing in agile software teams," *Inf. Syst. J.*, vol. 26, no. 2, pp. 95–125, 2016.
 - [18] F. Anwer, S. Aftab, S. Shah Muhammad Shah, U. Waheed, S. M. Shah, and U. Waheed, "Comparative analysis of two popular agile process models: extreme programming and scrum," *Int. J. Comput. Sci. Telecommun.*, vol. 8, no. 2, pp. 1–7, 2017.
 - [19] K. A. Seputra and G. Sandiasa, "Rancang Bangun Sistem Informasi Satgas Gotong Royong (Si Garong) Desa Adat Berbasis Mobile," *J. Nas. Pendidik. Tek. Inform.*, vol. 9, no. 3, p. 338, 2020.
 - [20] M. Larassati, A. Latukolan, A. Arwan, and M. T. Ananta, "Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram Ke Dalam Database," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, pp. 4058–4065, 2019.
 - [21] A. Bangor, T. Staff, P. Kortum, J. Miller, and T. Staff, "Determining what individual SUS scores mean: adding an adjective rating scale," *J. usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.