

PERLINDUNGAN STORAN MELALUI BERBILANG AWAN DENGAN PENGHIRISAN DATA

Avinesh Leo Adrian Fernandez

Elankovan A Sundararajan

Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia

ABSTRAK

Pengkomputeran awan telah menjadi semakin tersedia sejak akhir-akhir ini. Ini telah membolehkan individu dan perniagaan beralih kepada menyimpan data mereka di awan di mana ruang penyimpanan dan ketersediaan data tidak menjadi masalah. Keselamatan, manakala, adalah salah satu bidang yang menjadi perhatian utama terutama bagi syarikat, perbadanan, mahupun individu yang menyimpan data sensitif di awan. Walaupun data disimpan dengan selamat di storan tempatan, setelah data berada di awan, pemilik data tidak mempunyai kuasa untuk mengawalinya. Objektif penyelidikan ini adalah untuk mencari cara penyimpanan data di awan dengan selamat sementara masih tersedia apabila diperlukan. Penyelesaian yang dicadangkan untuk masalah ini adalah sebuah alat untuk menghiris, menyulitkan, dan memuat naik fail secara automatik ke penyedia perkhidmatan awan yang berlainan. Secara amnya, alat ini menghiris data kepada beberapa kepingan yang kecil. Setiap kepingan data kemudiannya disulitkan menggunakan algoritma AES sebelum masing-masing dimuat naik ke dua penyedia perkhidmatan awan yang berbeza. Apabila pengguna ingin mengakses data tersebut, alat tersebut akan memuat turun kepingan data yang berikut dan menyahsulitkan data sebelum membina semula fail asal dan disediakan untuk pengguna.

1 PENGENALAN

Pengkomputeran awan telah merevolusikan cara kita menggunakan komputer dan Internet pada zaman ini. Banyak aplikasi mudah alih dan web kini menggunakan pengkomputeran awan untuk menyimpan data masing-masing. Pengkomputeran awan boleh dikategorikan kepada Awan Persendirian (*Private Cloud*), Awan Hibrid (*Hybrid Cloud*), Awan Awam (*Public Cloud*), dan Awan Komuniti (*Community Cloud*). Setiap satu kategori awan ini mempunyai tahap keselamatan, pematuhan dasar dan bidang kuasa yang berbeza. Sebagai contoh, Awan Hibrid merupakan kombinasi awan awam dan persendirian di mana organisasi dapat menyimpan data sensitif dalam storan awan persendirian dan selebihnya disimpan dalam storan awan awam yang juga boleh digunakan oleh sesiapa sahaja (Suzita Awaluddin 2015).

Penggunaan pengkomputeran awan membolehkan pengguna mengakses maklumat dari mana-mana lokasi sekitar dunia. Antara contoh data yang sering disimpan di awan seperti gambar, video, dokumen, dan audio. Perkhidmatan awan telah banyak berkembang sejak hari-hari awalnya dan kini menawarkan kapasiti storan yang sangat besar dan mudah disediakan. Ini telah membuka ruang kepada banyak syarikat dan perbadanan besar yang sebelum ini menyimpan jumlah data yang besar dalam server tempatan, memindah kesemua data mereka kepada storan awan. Dengan ini, masalah kerosakan cakera keras yang mengakibatkan kehilangan data yang sering dihadapi oleh syarikat dan perbadanan sebegini dapat dielakkan. Antara kelebihan penyimpanan data di awan yang lain seperti penjimatan kos, skalabiliti dan kebolehcapaian.

Pengkomputeran awan bukan sahaja terhad kepada penyimpanan data tetapi juga menawarkan perkhidmatan yang lain seperti Infrastruktur sebagai Perkhidmatan *IaaS* (*Infrastructure-as-a-Service*), Platform sebagai Perkhidmatan *PaaS* (*Platform-as-a-Service*), dan Perisian Sebagai Perkhidmatan *SaaS* (*Software-as-a-Service*).

2 PENYATAAN MASALAH

Perkembangan pengkomputeran awan berjaya meyakinkan ramai individu, syarikat, dan perbadanan untuk menggunakan perkhidmatan awan yang ditawarkan. Namun, keselamatan dan integriti data bukan lagi di tangan pengguna selepas dimuat naik ke storan awan. Walaupun proses pengesahan dilaksanakan, sekiranya storan awan berjaya diceroboh, penceroboh akan dapat mengakses apa-apa maklumat yang disimpan di storan awan. Bukan itu sahaja, potensi pekerja di pihak penyedia perkhidmatan awam mengusik dan mengganggu data dengan niat yang tidak baik (*rogue employee*) harus dipertimbangkan. Kebocoran data sensitif akibat serangan siber juga menjadi kerisauan.

3 OBJEKTIF KAJIAN

Kajian ini dilakukan dengan matlamat memastikan data yang disimpan di storan awan awam selamat serta mencegah sebarang kebocoran data sensitif akibat serangan pencerobohan.

Bagi mencapai matlamat ini, objektif berikut digariskan:-

- 3.1 Mengenal pasti teknik penghirisan dan penyulitan data yang bersesuaian.
- 3.2 Mereka bentuk sebuah perisian yang dapat menghiris data, menyulit kepingan data, merawakkan kepingan data dan memuat naik kepingan data ke storan awam yang berlainan.
- 3.3 Mereka bentuk sebuah perisian yang dapat memuat turun kepingan data, menyahsulitkan kepingan data, dan membina semula fail asal.
- 3.4 Membangunkan perisian tersebut.

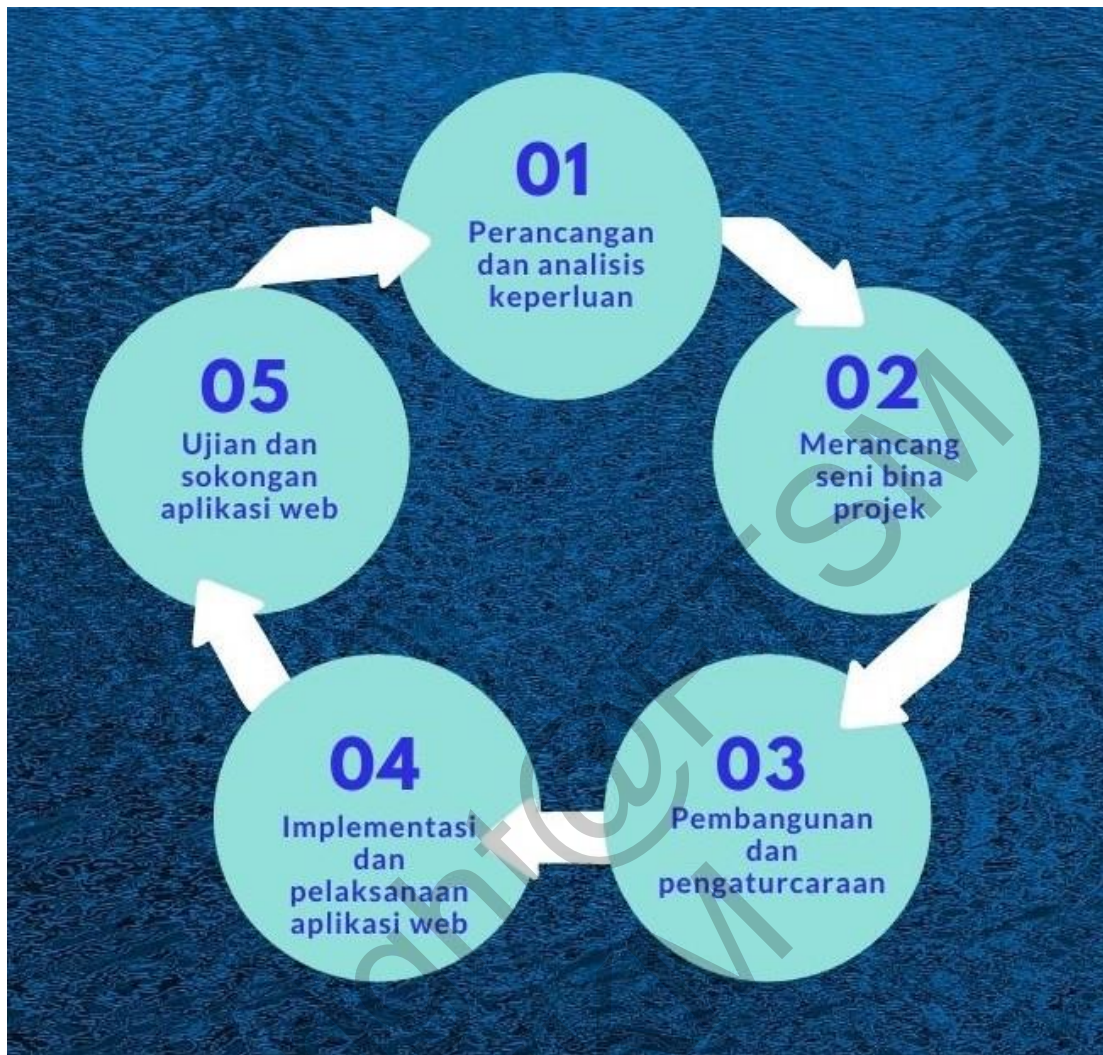
4 METOD KAJIAN

Pembangunan alatan ini boleh dibahagikan kepada beberapa fasa. Model Kitar Hayat dipilih sebagai metodologi untuk pembangunan aplikasi ini. Model ini dipilih kerana merupakan model yang paling senang diikuti, mudah difahami serta kurang risikonya.

Lima fasa utama Model Kitaran Hayat Pembangunan seperti berikut:

4.1 Fasa Perancangan

Fasa ini bermula dengan pengenalan masalah, objektif, skop dan kekangan kajian. Kajian kesusasteraan diadakan untuk mencetus idea serta menjadi inspirasi atau rujukan. Penggunaan internet untuk mencari bahan rujukan dan maklumat dikumpul. Penganalisan dan tafsiran atas maklumat terkumpul. Keperluan produk akhir ditetapkan hasil fasa ini.



Rajah 1.1: Model Kitar Hayat Pembangunan

4.2 Fasa Reka Bentuk

Fasa ini melibatkan dua proses penting iaitu mereka bentuk fungsi dan pengantar muka aplikasi. Mock-up pengantar muka disediakan. Garis panduan fungsi aplikasi menggunakan API awan disediakan.

Pengguna sistem ini merupakan sesiapa sahaja yang menginginkan fail mereka disimpan dengan selamat dalam storan awan. Dengan sistem ini, pengguna dapat menyimpan fail dalam storan awan dengan selamat. Keperluan pengguna dan keperluan fungsian seperti berikut:

a) Pendaftaran Pengguna dan Log Masuk Akaun

Aplikasi harus membenarkan pengguna baharu mencipta akaun dan log masuk ke dalam akaun yang dicipta.

b) Log Keluar Akaun

Aplikasi harus membenarkan pengguna log keluar dari akaun selepas tamat penggunaan.

c) Memuat Naik Fail

Aplikasi harus membenarkan pengguna memuat naik fail ke dalam storan awan.

Aplikasi harus membenarkan pengguna memilih saiz kepingan data atau jumlah kepingan data yang digunakan semasa proses penghirisan data.

d) Memuat Turun Fail

Aplikasi harus membenarkan pengguna memuat turun fail yang disimpan dalam storan awan.

Keperluan fungsian merupakan tingkah laku asas sesebuah sistem. Fungsi asas aplikasi ini dikemukakan supaya proses implementasi dapat dilaksanakan dengan lancar dan sempurna.

a) Penghirisan Data (*Data Slicing*)

Aplikasi membahagikan fail yang dipilih kepada beberapa kepingan. Aplikasi membahagikan fail mengikut ketentuan pengguna.

b) Penyulitan Data (*Encryption*)

Aplikasi melaksanakan proses penyulitan atas kepingan data.

c) Perawakkan Data (*Data Shuffling*)

Aplikasi menyusun semula kepingan data secara rawak.

d) Pemuatan Naik dan Turun Data (*Upload and Download*)

Aplikasi menggunakan API (*Application Programming Interface*) perkhidmatan storan awan untuk memuat naik dan turun data kepada dan daripada storan awan AWS dan Google Cloud.

e) **Penyahsulitan Data (*Decryption*)**

Aplikasi menyahsulitkan data yang dimuat turun dari storan awan.

f) **Pembinaan Semula Data (*Data Assembly*)**

Aplikasi membina semula fail asal dengan melengkapkan kepingan data.

Keperluan bukan fungsian juga penting dari segi menyokong keperluan fungsian yang diberi. Berikut adalah keperluan bukan fungsian dalam aplikasi ini:-

a) **Keselamatan**

1. Aplikasi perlu melindungi fail pengguna
2. Aplikasi perlu menyimpan maklumat pengguna dengan selamat
3. Aplikasi tidak membenarkan pengguna yang tidak berdaftar log masuk ke dalam sistem

b) **Kegunaan**

1. Aplikasi perlu mempunyai antara muka yang senang difahami dan digunakan
2. Aplikasi perlu menggunakan aturan elemen yang sesuai dan berterus terang
3. Pengguna dapat memahami cara penggunaan dari kali pertama menggunakannya

c) **Kecekapan**

1. Aplikasi perlu digunakan dengan lancar
2. Aplikasi perlu dilancarkan dengan pantas berdasarkan kelajuan Internet

d) Kebolehpercayaan

1. Aplikasi perlu berfungsi dengan stabil pada bila-bila masa
2. Aplikasi perlu bebas daripada sebarang ralat teknikal

e) Ketersediaan

1. Aplikasi perlu dicapai dari mana-mana lokasi menggunakan Internet
2. Aplikasi perlu membenarkan pengguna log masuk menggunakan mana-mana komputer

Keperluan merangkumi spesifikasi pembangun serta pengguna yang disyorkan.

a) Pembangunan

Keperluan perkakasan pembangunan aplikasi adalah seperti berikut:-

- Komputer: OS Windows 7 (64-bit) atau ke atas
- CPU Intel Core i5 atau ke atas
- 8GB RAM atau ke atas
- Rangkaian Internet (10Mbps atau ke atas)

Keperluan perisian pembangunan aplikasi adalah seperti berikut:-

- Eclipse IDE
- Java JDK 15
- API: AWS dan Google Cloud

b) Penggunaan

Keperluan perkakasan penggunaan aplikasi adalah seperti berikut:-

- Komputer: OS Windows 7 (64-bit) atau ke atas
- 4GB RAM atau lebih
- Rangkaian Internet (10Mbps atau ke atas)

4.3 Fasa Pembangunan

Fasa ini bermula dengan perkembangan aplikasi. Seterusnya, pengaturcaraan dimulakan diikuti dengan penyusunan dan pengujian.

Pembangunan projek ini menggunakan kerangka pengaturcaraan Java with Maven. Penggunaan Java Maven bermaksud *dependencies* yang diperlukan dapat disertakan dalam projek dengan pantas. Antara *dependencies* penting yang digunakan dalam projek ini ialah `google-cloud-storage` yang diperlukan untuk menggunakan API mengakses bucket, memuat naik, dan turun object dari Google Cloud Storage. Selain itu, *dependency* `aws-java-sdk` juga digunakan untuk mengakses bucket dan object dari AWS S3.

API perkhidmatan awan Google Cloud Storage dan Amazon AWS S3 yang digunakan boleh didapati daripada dokumentasi yang disediakan oleh penyedia awan masing-masing. Dokumentasi Google Cloud Storage dan Amazon AWS S3 menyediakan segmen kod Java yang diperlukan oleh modul muat naik dan turun.

Berikut merupakan antara API yang diperoleh daripada dokumentasi perkhidmatan awan dan diubah suai mengikut keperluan projek.

a) Muat Naik

```
11 public class UploadObjectGCS {
12     public static void uploadObject(String objectName, String filePath) throws IOException {
13         // The ID of your GCP project
14         String projectId = AccountDetails.getGcsProjectID();
15
16         // The ID of your GCS bucket
17         String bucketName = AccountDetails.getGcsBucketName();
18
19         // The ID of your GCS object
20         // String objectName = "your-object-name";
21
22         // The path to your file to upload
23         // String filePath = "path/to/your/file"
24
25         Storage storage = StorageOptions.newBuilder().setProjectId(projectId).build().getService();
26         BlobId blobId = BlobId.of(bucketName, objectName);
27         BlobInfo blobInfo = BlobInfo.newBuilder(blobId).build();
28         storage.create(blobInfo, Files.readAllBytes(Paths.get(filePath)));
29
30         System.out.println(
31             "File " + filePath + " uploaded to bucket " + bucketName + " as " + objectName);
32     }
}
```

Gambar 4.1: Segmen Kod Kelas UploadObject GCS


```

9 public class UploadObjectAWS {
10     public static void uploadObject(String keyName, String filePath) {
11
12         String bucket_name = AccountDetails.getAwsBucketName();
13         String file_path = filePath;
14         String key_name = keyName;
15
16         System.out.format("Uploading %s to S3 bucket %s...\n", file_path, bucket_name);
17         final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withRegion("ap-southeast-1").build();
18         try {
19             s3.putObject(bucket_name, key_name, new File(file_path));
20         } catch (AmazonServiceException e) {
21             System.err.println(e.getErrorMessage());
22             System.exit(1);
23         }
24         //System.out.println("Done!");
25     }
26 }

```

Gambar 4.2: Segmen Kod Kelas UploadObject AWS

Pemboleh ubah *projectID* dan *bucketName* diperlukan oleh perisian untuk mengakses storan pengguna. *filePath* pula merupakan lokasi fail yang ingin dimuat naik oleh pengguna.

Kod yang digunakan dalam modul penghirisan dan pembinaan semula diperoleh daripada laman web StackOverflow hasil sumbangan pengguna dengan nama pengguna Pshemo.

Kelas Java *FileSplit* yang diberi mempunyai tiga *method* yang masing-masing berfungsi membahagi fail kepada kepingan kecil, menyimpan dan menyusun kepingan fail dalam sebuah *ArrayList*, dan membina semula fail asal.

```

14 public class FileSplit {
15     public static List<?> splitFile(File f, double sliceSize, String path) throws IOException {
16         int partCounter = 1; //I like to name parts from 001, 002, 003, ...
17         //you can change it to 0 if you want 000, 001, ...
18
19         int sizeOfFiles = (int) (1024 * 1024 * sliceSize); // 1MB
20         byte[] buffer = new byte[sizeOfFiles];
21         List<String> fileOrder = new ArrayList<>();
22
23         String fileName = f.getName();
24
25         //try-with-resources to ensure closing stream
26         try (FileInputStream fis = new FileInputStream(f);
27             BufferedInputStream bis = new BufferedInputStream(fis)) {
28
29             int bytesAmount = 0;
30             while ((bytesAmount = bis.read(buffer)) > 0) {
31                 //write each chunk of data into separate file with different number in name
32                 String filePartName = String.format("%s.%03d", fileName, partCounter++);
33                 File newFile = new File(path, filePartName);
34                 try (FileOutputStream out = new FileOutputStream(newFile)) {
35                     out.write(buffer, 0, bytesAmount);
36                 }
37             }
38         }
39     }
40 }

```

Gambar 4.5: Segmen Kod Kelas FileSplit 1

Method `splitFile` membahagikan fail kepada beberapa kepingan kecil mengikut saiz maksimum yang telah ditentukan oleh pengguna. *ByteArray* digunakan untuk memastikan kepingan data yang dihasilkan tidak melebihi saiz yang ditentukan.

```

45● public static void mergeFiles(List<File> files, File into)
46     throws IOException {
47     try (FileOutputStream fos = new FileOutputStream(into);
48         BufferedOutputStream mergingStream = new BufferedOutputStream(fos)) {
49         for (File f : files) {
50             Files.copy(f.toPath(), mergingStream);
51         }
52     }
53 }
54
55● public static List<File> listOfFilesToMerge(File oneOfFiles) {
56     String tmpName = oneOfFiles.getName(); //{name}.{number}
57     String destFileName = tmpName.substring(0, tmpName.lastIndexOf('.')); //remove .{number}
58     File[] files = oneOfFiles.getParentFile().listFiles(
59         (File dir, String name) -> name.matches(destFileName + "[.]\d+"));
60     Arrays.sort(files); //ensuring order 001, 002, ..., 010, ...
61     return Arrays.asList(files);
62 }

```

Gambar 4.6: Segmen Kod Kelas `FileSplit 2`

Method `listOfFilesToMerge` berfungsi menyimpan lokasi setiap kepingan fail dalam sebuah *ArrayList* dan menyusun kepingan mengikut urutan. *Method* `mergeFiles` berfungsi membina semula fail dengan mencantumkan kepingan fail yang disimpan dalam *ArrayList* yang tersebut di atas.

b) *Credentials* (Kelayakan)

Fungsi log masuk dan pendaftaran akaun dalam usulan projek tidak diteruskan dalam proses pembangunan. *Credentials* (Kelayakan) yang diperlukan untuk mengakses storan awan masing-masing seperti `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` untuk `AWS S3` dan `GOOGLE_APPLICATION_CREDENTIALS` untuk `GCS` lebih bersesuaian digunakan untuk tujuan keselamatan.

API storan awan `AWS S3` dan `GCS` masing-masing memerlukan kelayakan untuk membuat sambungan ke *bucket* storan awan pengguna. Sekiranya, kelayakan tidak diberikan, perisian tidak dapat mengakses storan awan dan fail tidak dapat dimuat naik atau turun. Setiap pengguna memiliki kelayakan yang berlainan dan hanya dapat mengakses *bucket* kepunyaan pengguna masing-masing. Pengguna haruslah menyimpan data kelayakan ini dengan selamat supaya dapat menggunakan aplikasi ini dengan sempurna. Kehilangan

mana – mana data kelayakan ini bermakna pengguna tidak dapat mengakses fail yang telah dimuat naik ke storan awan. Kebocoran data ini juga mengundang bahaya sekiranya terjatuh kepada pihak yang salah. Oleh itu, data kelayakan ini haruslah disimpan dengan selamat.

Pemboleh ubah persekitaran (*Environment Variable*) menyediakan cara untuk menentukan pilihan konfigurasi dan kelayakan. Pengguna harus membuat akaun perkhidmatan secara manual. Kemudian, pengguna akan dapat membuat satu atau lebih kunci akaun perkhidmatan, yang merupakan kelayakan yang berkaitan dengan akaun perkhidmatan. Kunci akaun perkhidmatan kemudian boleh ditetapkan sebagai pemboleh ubah persekitaran. Kaedah menetapkan pemboleh ubah boleh dirujuk kepada artikel bertajuk “*How To Set Environment Variables*” (Dominik Kundel).

Copyright@FTSM
UKM

4.4 Fasa Pengujian

Ini adalah fasa dimana aplikasi diuji berulang kali. Selepas itu, aplikasi yang direka bentuk akan dibangunkan untuk kegunaan pengguna.

Fasa pengujian antara fasa yang terpenting sepanjang proses pembangunan sesebuah aplikasi, perisian, sistem atau laman web. Fasa pengujian dapat membantu memastikan sesebuah perisian berfungsi dengan sempurna dan tiada sebarang masalah kritikal dihadapi oleh pengguna selepas perisian telah dipasarkan. Fasa pengujian juga membantu pembangun perisian mengesan sebarang kesalahan kecil yang terlepas pandang semasa proses pembangunan.

Perisian Perlindungan Storan ini diuji dari segi kebolehgunaan aspek fungsian dan bukan fungsian. Keutamaan diberikan kepada pengujian aspek fungsian seperti penghantaran data, penyulitan, perawakan, dan pemuatan naik dan turun. Aliran data dan interaksi perisian dengan pengguna dan perkhidmatan storan awan juga turut diuji.

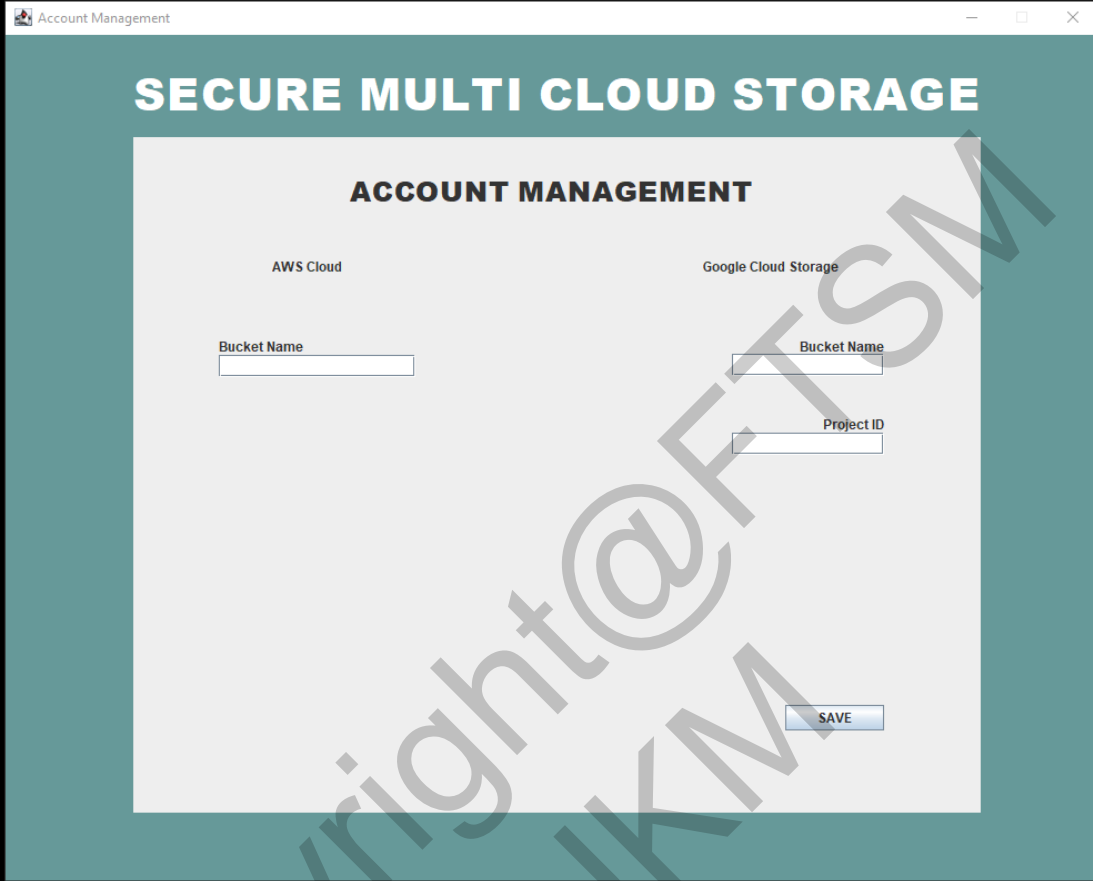
Ujian yang diperlukan untuk menguji keperluan fungsian perisian ialah *BlackBox Testing*.

5 HASIL KAJIAN

Bahagian ini membincangkan proses pembangunan perisian Perlindungan Storan Melalui Berbilang Awan dengan Penghantaran Data. Proses pembangunan perisian merangkumi modul, komponen, librari, dan pangkalan data yang digunakan dalam perisian ini. Selain itu, segmen kod komponen utama dikemukakan dalam dokumen ini. Masalah yang dihadapi sepanjang proses pembangunan dibincang dalam kesimpulan dokumen ini.

5.1 Antara Muka Pengguna

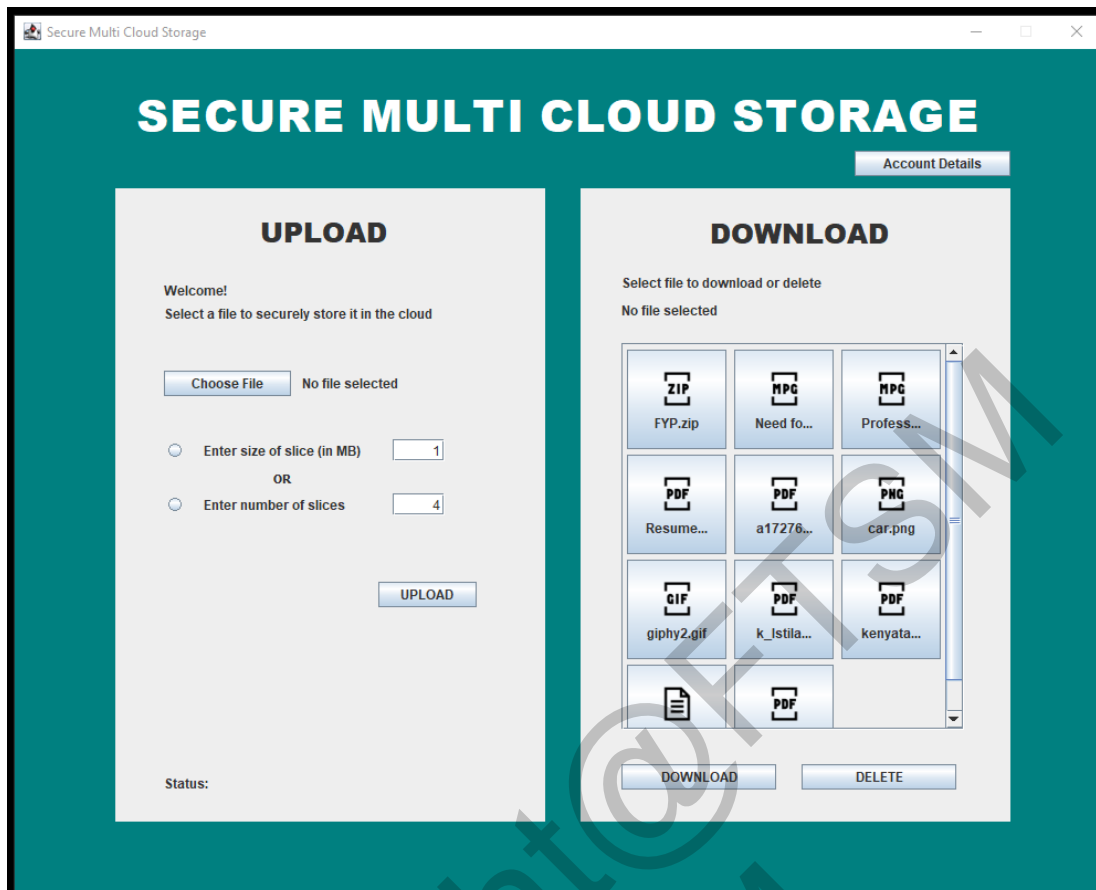
Perisian ini mempunyai dua antara muka iaitu skrin “*Account Management*” dan skrin “*Upload and Download*”. Skrin “*Account Management*” dipaparkan setiap kali perisian dilancarkan.

The screenshot shows a web application window titled "Account Management". The main heading is "SECURE MULTI CLOUD STORAGE" in a teal banner. Below it, the section is titled "ACCOUNT MANAGEMENT". There are two columns of input fields. The left column is for "AWS Cloud" and contains a "Bucket Name" input field. The right column is for "Google Cloud Storage" and contains a "Bucket Name" input field and a "Project ID" input field. A "SAVE" button is located at the bottom right of the form area. A large watermark "Copyright@FTSM" is overlaid diagonally across the image.

Gambar 4.10: Tangkapan Skrin Antara Muka Account Details

Terdapat tiga ruang teks yang harus diisi dalam skrin ini. Pada sebelah kiri, nama *bucket* AWS S3 pengguna harus diisi. Di sebelah kanan pula, nama *bucket* dan ID projek Google Cloud Storage pengguna harus diisi. Butang “*SAVE*” terletak di sudut kanan bawah berfungsi menyimpan data yang diisi dan memaparkan skrin seterusnya.

Seterusnya, skrin “*Account Details*” akan tersembunyi dan skrin “*Upload and Download*” akan dipaparkan.



Gambar 4.12: Tangkapan Skrin Antara Muka “Upload and Download”

Skrin “*Upload and Download*” dibahagikan kepada dua panel dengan fungsi muat naik pada sebelah kiri dan muat turun pada sebelah kanan.

5.2 Proses Muat Naik

Pada bahagian muat naik, terdapat butang “*Choose File*” untuk melancarkan *JFileChooser()* yang membuka peneroka fail sistem membenarkan pengguna memilih fail yang ingin dimuat naik. Pengguna juga dapat memilih kaedah penghirisan sama ada melalui saiz setiap kepingan ataupun jumlah kepingan.

Akhirnya, butang “*UPLOAD*” berfungsi memulakan sebuah *thread* proses yang akan memanggil modul penghirisan, penyulitan, perawakkan, dan memuat naik kepingan data ke storan awan. *Thread* digunakan kerana pemanggilan modul yang banyak menyebabkan antaramuka pengguna membeku sementara proses setiap modul dilaksanakan. Hal ini menjadi masalah yang lebih besar apabila fail yang bersaiz tinggi dipilih.

5.3 Proses Muat Turun

Pada bahagian muat turun, terletak *JScrollPane* yang membenarkan pengguna melihat fail yang telah dimuat naik ke storan awan. Setiap fail diwakili oleh sebuah *JButton* yang tertera nama fail dan ikon berdasarkan format fail masing-masing. Pengguna hanya dibenarkan memilih satu butang pada satu masa untuk tujuan muat turun ataupun memadam. Nama fail yang dipilih pengguna akan diterterakan di atas *JScrollPane* tersebut supaya pengguna sedar akan pilihan mereka. Butang “*DOWNLOAD*” digunakan untuk memuat turun fail selepas dipilih oleh pengguna. Butang “*DELETE*” pula digunakan untuk memadam fail yang dipilih pengguna dari storan awan selepas mendapat pengesahan.

6 KESIMPULAN

Kajian kesusasteraan yang diadakan membantu menjana idea dan memberi pengetahuan yang luas berkaitan tentang bidang dan topik Perlindungan Storan, Storan Awan, Keselamatan Data, dan Penghirisan Data. Metodologi dan Reka Bentuk merangkumi keperluan fungsian dan bukan fungsian perisian, definisi keperluan pengguna, dan spesifikasi keperluan perkakasan dan perisian, reka bentuk antara muka, pangkalan data, dan seni bina perisian. Bab pembangunan dan pengujian mendokumentasi proses pembangunan perisian seperti kod-kod setiap modul dan antara muka pengguna, serta hasil ujian perisian tersebut.

Antara batasan yang paling ketara ialah perisian ini hanya boleh digunakan dengan OS (*Operating System*) Windows. Pengguna telefon pintar yang menggunakan OS Android dan iOS tidak dapat menggunakan aplikasi ini. Pengguna MacOS dan Linux juga tidak dapat menggunakan perisian ini.

Batasan seterusnya ialah perisian hanya menggunakan dua penyedia perkhidmatan awan atas sebab kesukaran pembangunan dengan skop masa yang terhad. Penggunaan tiga atau lebih penyedia perkhidmatan awan akan meninggikan tahap kesukaran pembangunan dan mengambil masa yang lebih lama untuk memastikan kesemua API berfungsi dengan baik dan lancar.

Saiz maksimum fail yang dapat dimuat naik ialah 2GB dan juga bergantung kepada kelajuan muat naik Internet. Cadangan penambahbaikan yang utama ialah untuk mengembang perisian ini supaya dapat digunakan dengan sistem operasi lain yang banyak digunakan seperti Android, iOS, dan MacOS. Dengan ini, lebih ramai pengguna akan dapat menggunakan aplikasi ini dan akses fail dan data mereka menggunakan peranti yang berlainan.

Seterusnya, menggunakan lebih daripada dua penyedia perkhidmatan awan. Penggunaan penyedia perkhidmatan awan yang lebih dapat meningkatkan keselamatan data dengan mengasingkan data kepada lebih daripada dua storan awan supaya lebih sukar berlakunya pembocoran data. Contoh perkhidmatan awan lain yang boleh digunakan seperti Microsoft Azure dan IBM Cloud Object Storage.

7 RUJUKAN

- Andrew Thompson, 2012. How to break a file into pieces using Java?. [online] Stack Overflow. <https://stackoverflow.com/a/10864346>
- Balasaraswathi, V.R. and Manikandan, S., 2014, May. Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach. In 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp. 1190-1194). IEEE.
- Baiju NT., 2021. 5 advantages and disadvantages of Cloud Storage. Big Data Made Simple. <https://bigdata-madesimple.com/5-advantages-and-disadvantages-of-cloud-storage/>
- Google Cloud. n.d. Cloud Storage documentation | Google Cloud. <https://cloud.google.com/storage/docs/>
- Docs.aws.amazon.com. 2021. AWS SDK for Java Documentation. <https://docs.aws.amazon.com/sdk-for-java/index.html>
- Dominik K., 2017. How To Set Environment Variables. Twilio Blog. <https://www.twilio.com/blog/2017/01/how-to-set-environment-variables.html>
- EncodedBunny, 2017, File-Fragmenter: A Java API for fragmenting/splitting files. GitHub. <https://github.com/EncodedBunny/File-Fragmenter>
- GeeksforGeeks. 2018. Collections.shuffle() in Java with Examples - GeeksforGeeks. <https://www.geeksforgeeks.org/collections-shuffle-java-examples/>
- GitHub. 2019. googleapis/google-cloud-java. <https://github.com/googleapis/google-cloud-java>
- Gulshan Kumar. and Laxmi, V., An Approach for Securing Data on Cloud Using Data Slicing and Cryptography.
- How It Works. 25 April 2019. Cloud storage: What is it and how does it work? <https://www.howitworksdaily.com/cloud-storage-what-is-it-and-how-does-it-work/>
- IBM, IBM Cloud Education. 29 October 2019. PaaS (Platform-as-a-Service). <https://www.ibm.com/cloud/learn/paas>

- Junghanns P, Fabian B, Ermakova T. Engineering of secure multi-cloud storage, Institute of Information Systems, Spandauer Berlin Germany. 16 January 2016. Computers in industry 83 (2016) 108-120.
- Krishnasamy, K.Subramanian & John, Leo. 2017. Dynamic and secure unstructured data sharing in multi-cloud storage using the hybrid crypto-system. International Journal of ADVANCED AND APPLIED SCIENCES. 5. 10.21833/ijaas.2018.01.003.
- Mahdi Aza's, Elankovan Sundararajan. Performance Improvement For Cryptography, Compression And Data Transfer In Multi-Cloud Storage Through Parallelization. Research Centre for Software Technology and Management, 2020. PS-FTSM-2020-021.
- Nadeem, M. 2016 "Cloud Computing: Security Issues and Challenges", Journal of Wireless Communications, 1(1), pp. 10-15. doi: 10.21174/jowc.v1i1.73.
- Peng Xu, Xiaqi Liu. SSDS-MC: Slice-based Secure Data Storage in Multi-Cloud Environment, Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory. 2015.
- Rajpreet Kaur, & Singh, Paramjeet & Rani, Shaveta. (2018). Migrated Encrypted Data in Cloud using Data Slicing Approach. International Journal of Computer Sciences and Engineering. 6. 286-290. 10.26438/ijcse/v6i7.286290.
- Rupesh R Bobde, Amit Khaparde, Raghuwanshi. An approach for securing data on cloud using data slicing and cryptography, IEEE Sponsored 9th International Conference on Intelligent System and Control (ISCO) 2015.
- Sandeep Nehe, Vaidya. Data security using data slicing over storage clouds, International Conference on Information Processing (ICIP) Vishwakarma Institute of Technology. 19 December 2015.
- Shahid, F., Ashraf, H., Ghani, A., Ghayyur, S.A.K., Shamshirband, S. and Salwana, E., 2020. PSDS–Proficient Security Over Distributed Storage: A Method for Data Transmission in Cloud. IEEE Access, 8, pp.118285-118298.
- Svitlana Varaksina. 2019. Why Are Requirements Important in Software Engineering? Why Are Requirements Important in Software Engineering? - Mind Studios (themindstudios.com) [21/11/2020]

Wadah ICT UKM, Suzita. Cloud computing. 19 November 2015.
<http://www.ukm.my/wadahict/cloud-computing/>

Welkin X., 2017. WelkinXie/AESCipher-Java: AES encryption working between Objective-C and Java [GitHub]. <https://github.com/WelkinXie/AESCipher-Java>

www.javatpoint.com. n.d. Software Testing | Black Box Testing - javatpoint.
<https://www.javatpoint.com/black-box-testing>

Copyright@FTSM
UKM