

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi

Implementasi dan pengujian pada bab ini bertujuan untuk mengetahui tahapan implementasi pada prototype sistem yang dibuat. Tahapan ini dilakukan setelah analisis dan perancangan sistem selesai dilakukan. setelah pembangunan sistem selesai dilakukan implementasi sistem yang terdiri dari implementasi perangkat keras, implementasi perangkat lunak dan implementasi basis data dan pengujian

##### 4.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan perangkat keras yang digunakan untuk mengimplementasikan sistem yang dibuat. Perangkat ini berupa *web hosting* yang disewakan ke jasa hosting dengan spesifikasi yang ditampilkan pada tabel 4.1 berikut.

Tabel 4.1 Spesifikasi Perangkat Keras Implementasi Sistem

No	Perangkat Keras	Spesifikasi
1.	SSD Storage	10 GB
2.	CPU / Processor	4.4 GHz
3.	RAM	8 GB
4.	VGA	1.8 GHz

##### 4.1.2 Implementasi Perangkat Lunak

Berikut ini adalah implementasi perangkat lunak yang digunakan dalam pembangunan sistem yang dibuat ditampilkan pada tabel 4.2 berikut.

Tabel 4.2 Spesifikasi Perangkat Lunak Implementasi Sistem

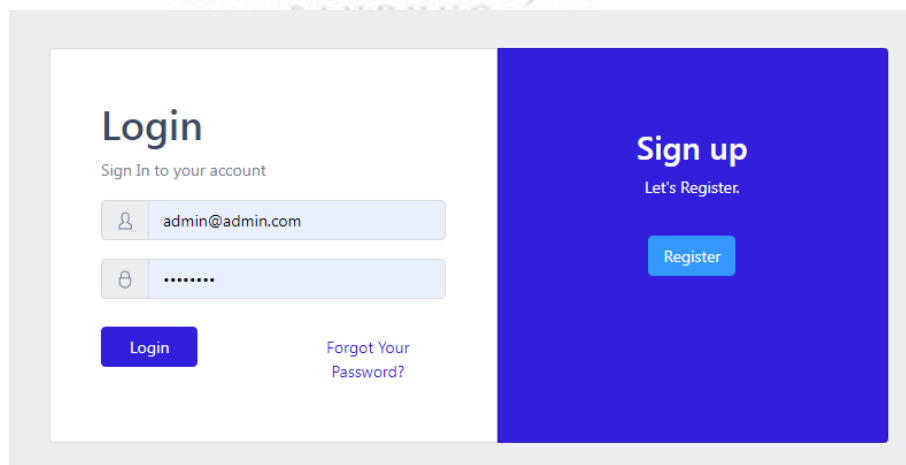
No	Perangkat Lunak	Spesifikasi
1	OS	Ubuntu Server Versi 18
2	Mysql	5.0.67
3	Apache	2.4.34
4	PHP	8.1

#### 4.1.3 Implementasi User Interfaces

User interface yang ada diimplementasikan adalah hasil rancangan dari mockup yang sudah dibuat sebelumnya. User interface yang ada dalam aplikasi ini meliputi halaman login, halaman registrasi, halaman dashboard, halaman file storage, dan halaman upload file. Berikut ini adalah penjelasan masing-masing halaman pada aplikasi.

##### 1. Halaman Login

Halaman login adalah halaman pertama kali muncul ketika aplikasi diakses. Halaman ini bertujuan untuk memberikan keamanan pada aplikasi agar tidak diakses oleh user yang tidak memiliki hak akses. Dalam melakukan akses ke sistem, pengguna perlu memasukan alamat email yang terdaftar dan memasukan password yang valid. Proses validasi yang bekerja pada bagian ini akan menentukan apakah email yang dimasukan terdaftar di sistem atau tidak. Jika email terdaftar dalam sistem, maka user akan diarahkan ke halaman dashboard sistem. Adapun tampilan dari implementasi halaman ini ditunjukkan pada gambar 4.1 berikut ini.



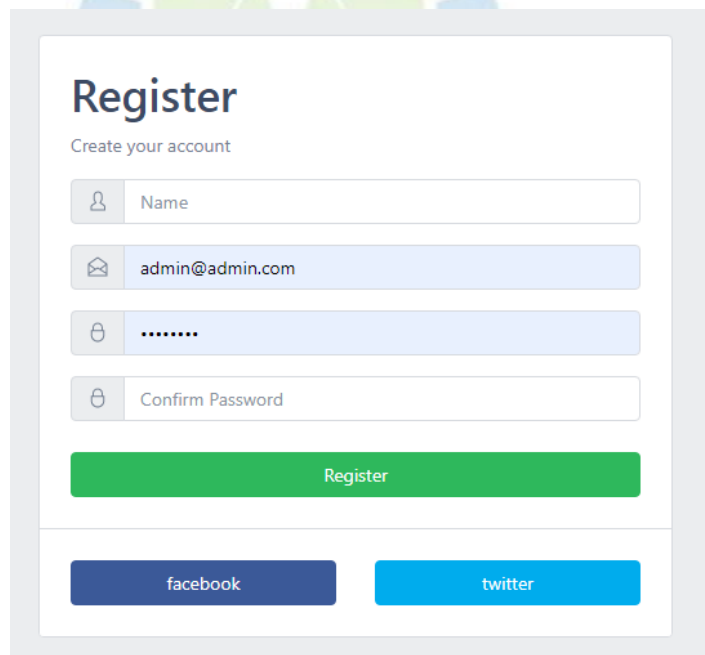
The image shows a user interface for a login and registration system. It is divided into two main sections: 'Login' and 'Sign up'. The 'Login' section on the left is white and contains the text 'Login' and 'Sign In to your account'. It features two input fields: one for email (containing 'admin@admin.com') and one for password (masked with dots). Below these fields are a blue 'Login' button and a link that says 'Forgot Your Password?'. The 'Sign up' section on the right is blue and contains the text 'Sign up' and 'Let's Register.'. It features a light blue 'Register' button.

Gambar 4.1 Halaman Login

Halaman login adalah halaman awal yang ditampilkan pada saat membuka halaman admin (*backend*) halaman berfungsi untuk admin atau mitra untuk masuk kedalam aplikasi web. Pada bagian tersebut terdapat beberapa bagian yaitu inputan email dan password. Pada sisi sebelah kanan terdapat bagian tombol registrasi yang berguna untuk user membuat akun baru.

## 2. Halaman Registrasi

Halaman kedua adalah halaman registrasi, yang berguna untuk user baru melakukan pendaftaran pada aplikasi. Adapun implementasi halaman registrasi pada sistem yang dibuat ditunjukkan oleh gambar 4.2. Pada halaman ini terdapat beberapa inputan yang perlu diisi oleh user baru, yaitu nama pengguna, email, password, dan konfirmasi password. Setelah pengguna mengisi semua kebutuhan data yang diperlukan, selanjutnya pengguna tinggal klik tombol “registrasi” untuk menyelesaikan alur pendaftaran ke akun.



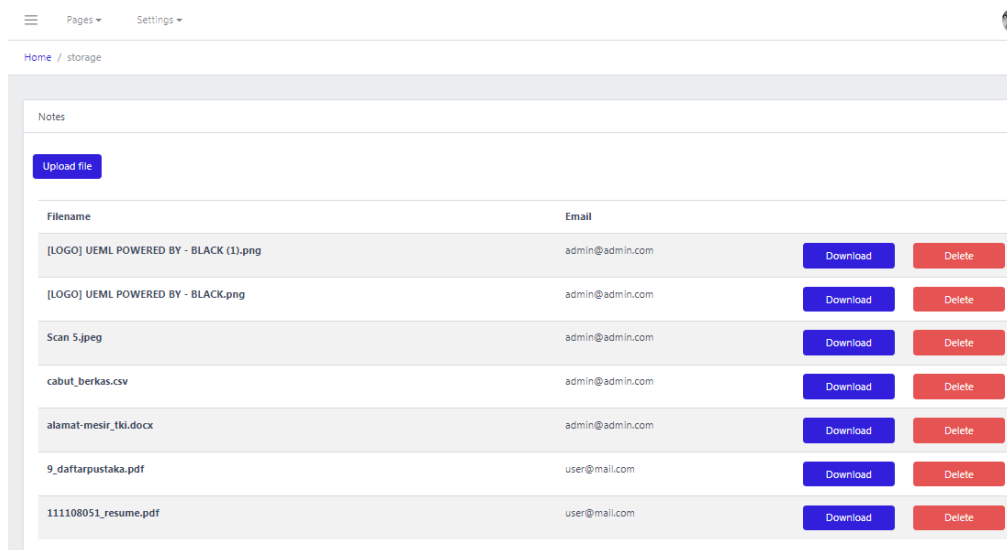
The image shows a web registration form titled "Register" with the subtitle "Create your account". The form is contained within a light gray border. It features four input fields, each with a small icon on the left: a person icon for "Name", an envelope icon for "Email" (containing the text "admin@admin.com"), a lock icon for "Password" (containing masked dots "\*\*\*\*\*"), and another lock icon for "Confirm Password". Below these fields is a prominent green button labeled "Register". At the bottom of the form, there are two additional buttons: a dark blue button labeled "facebook" and a light blue button labeled "twitter".

Gambar 4.2 Halaman Registrasi

## 3. Halaman File Storage

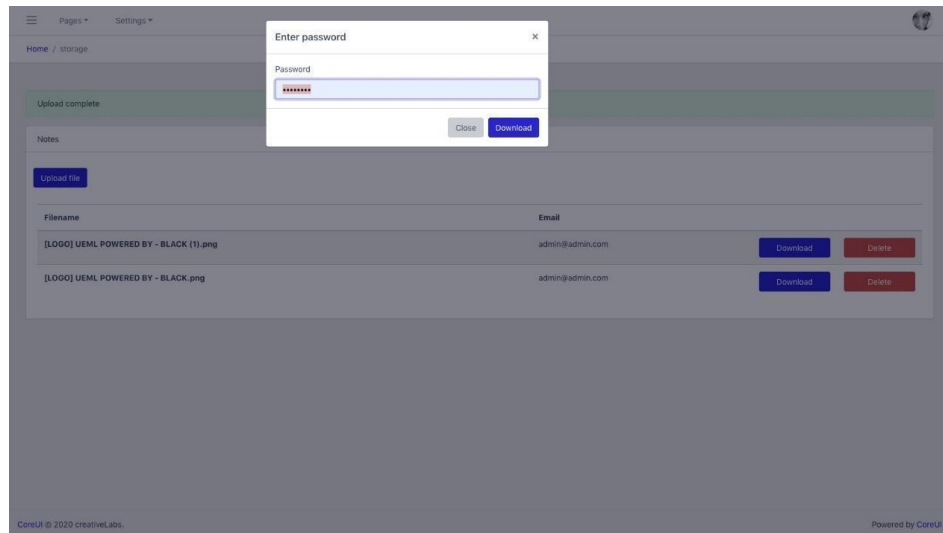
Halaman file storage berguna untuk menampilkan semua file yang sudah diupload ke cloud. Pada halaman file storage ini, ada dua bagian utama, yaitu *upload file* dan *list file*. Upload file berfungsi untuk

menampilkan halaman upload file. Sedangkan pada *list file* atau daftar file berfungsi menampilkan semua file-file yang sudah diupload ke *cloud*. Pada bagian list file terdapat bagian filename yang menunjukkan nama dari file, bagian email yang menunjukkan siapa yang melakukan proses upload, bagian tombol download untuk melakukan proses download dan bagian tombol delete untuk menghapus file. Adapun hasil dari implementasi halaman *file storage* ditunjukkan pada Gambar 4.3 dibawah ini.



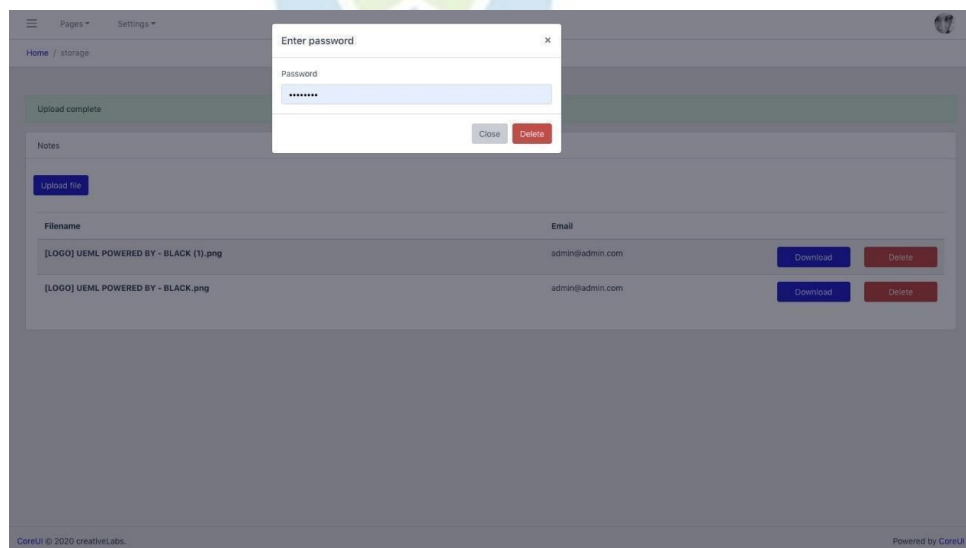
Gambar 4.3 Halaman File Storage

Halaman tersebut terdapat dua tombol, yaitu download dan delete. Pada saat user mengklik tombol download, maka sistem akan menampilkan password untuk diisi oleh pengguna (gambar 4.4). Jika password yang diinput oleh pengguna salah, maka proses download file tidak bisa dilakukan. Sebaliknya, jika password yang dimasukkan oleh pengguna benar. Maka sistem menjalankan proses download file.



Gambar 4.4 Input Password untuk download file

Proses yang sama juga dilakukan pada saat user melakukan penghapusan data. Dimana data yang akan dihapus harus diverifikasi dengan sebuah password (Gambar 4.5). Jika password yang diinput oleh pengguna benar, maka proses penghapusan data password bisa dilakukan.



Gambar 4.5 Proses Hapus Data

#### 4. Halaman Upload File

Halaman upload file pada sistem ini diimplementasikan seperti pada gambar 4.6 berikut ini.

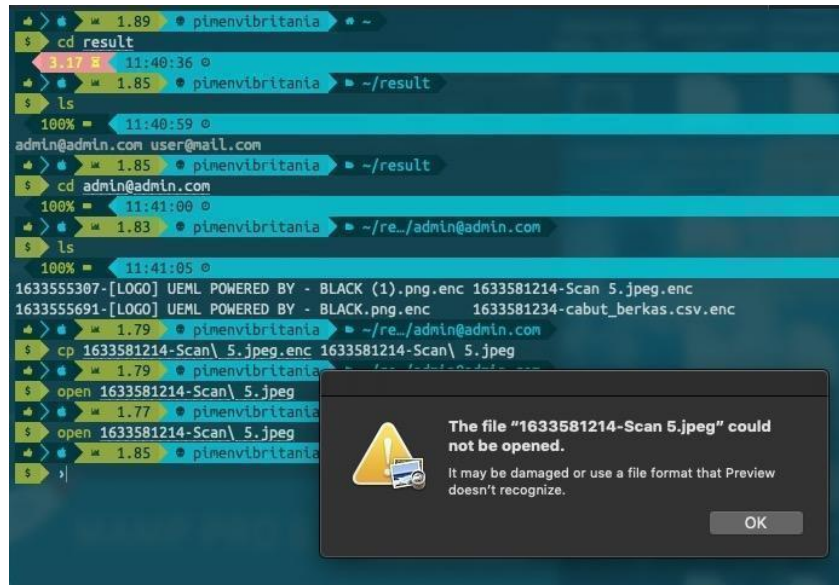
Gambar 4.6 Halaman Upload File

Halaman upload file digunakan sebagai user untuk mengupload file ke cloud storage. Pada bagian ini terdapat komponen upload file, password, tombol upload dan return. Dalam melakukan upload file, user diminta mengklik bagian upload file dan memilih file mana yang akan diupload. Setelah itu, pengguna diminta untuk mengisi password sesuai dengan password user kemudian mengklik tombol upload untuk menjalankan proses upload. Setelah proses upload selesai, maka file akan tersimpan ke cloud storage (pada penelitian ini menggunakan Amazon S3) seperti yang ditunjukkan pada gambar 4.7 berikut ini.

Gambar 4.7 Hasil upload file di Amazon S3

Setiap file yang di upload akan diberikan format baru berupa .env sebagai penanda bahwa file sudah dilakukan proses enkripsi dan kompresi dengan algoritma AES dan LZW. Penambahan proses ini memberikan keamanan kepada file agar tidak bisa dibuka oleh user lain. Jika user lain

mencoba membuka file tersebut, maka muncul *alert* dari komputer seperti gambar 4.8 dibawah.



Gambar 4.8 Notifikasi Sistem Error

#### 4.1.4 Implementasi Algoritma AES dan LZW

Implementasi algoritma AES dan LZW pada sistem enkripsi *file* di PT. Generasi Muda Indonesia Utama adalah sebagai berikut.

##### 4.1.4.1 Proses Enkripsi dengan Algoritma AES

Dalam algoritma AES, melakukan proses dekripsi perlu dilakukan beberapa proses transformasi diantaranya sBox, InvSubByte, InvShiftRow, InvMixColumn dan AddRoundKey. Berikut ini adalah coding-coding yang digunakan untuk melakukan proses enkripsi dengan algoritma AES.

###### a. Sbox

S-Box pada algoritma AES memiliki nilai baku yang telah ditetapkan dalam bentuk array 16 x 16 berupa angka hexadecimal, berikut ini adalah coding untuk S-Box.

```
var $sBox = array(  
    array(0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5,  
0x30, 0x01, 0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76),  
    array(0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0,
```

```

0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72, 0xC0),
    array(0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC,
0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15),
    array(0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A,
0x07, 0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2, 0x75),
    array(0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0,
0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84),
    array(0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B,
0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF),
    array(0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85,
0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C, 0x9F, 0xA8),
    array(0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5,
0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2),
    array(0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17,
0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19, 0x73),
    array(0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88,
0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B, 0xDB),
    array(0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C,
0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79),
    array(0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9,
0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE, 0x08),
    array(0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6,
0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x8A),
    array(0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E,
0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D, 0x9E),
    array(0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94,
0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF),
    array(0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68,
0x41, 0x99, 0x2D, 0x0F, 0xB0, 0x54, 0xBB, 0x16)
);

```

#### b. *Invsub*

Coding dibawah ini merupakan invers dari nilai s-box untuk menampilkan nilai dalam bentuk array.

```

// The inverse S-Box substitution table.
var $invSBox = array(
    array(0x52, 0x09, 0x6A, 0xD5, 0x30, 0x36, 0xA5, 0x38,
0xBF, 0x40, 0xA3, 0x9E, 0x81, 0xF3, 0xD7, 0xFB),

```



```

        array(0x7C, 0xE3, 0x39, 0x82, 0x9B, 0x2F, 0xFF, 0x87,
0x34, 0x8E, 0x43, 0x44, 0xC4, 0xDE, 0xE9, 0xCB),
        array(0x54, 0x7B, 0x94, 0x32, 0xA6, 0xC2, 0x23, 0x3D,
0xEE, 0x4C, 0x95, 0x0B, 0x42, 0xFA, 0xC3, 0x4E),
        array(0x08, 0x2E, 0xA1, 0x66, 0x28, 0xD9, 0x24, 0xB2,
0x76, 0x5B, 0xA2, 0x49, 0x6D, 0x8B, 0xD1, 0x25),
        array(0x72, 0xF8, 0xF6, 0x64, 0x86, 0x68, 0x98, 0x16,
0xD4, 0xA4, 0x5C, 0xCC, 0x5D, 0x65, 0xB6, 0x92),
        array(0x6C, 0x70, 0x48, 0x50, 0xFD, 0xED, 0xB9, 0xDA,
0x5E, 0x15, 0x46, 0x57, 0xA7, 0x8D, 0x9D, 0x84),
        array(0x90, 0xD8, 0xAB, 0x00, 0x8C, 0xBC, 0xD3, 0x0A,
0xF7, 0xE4, 0x58, 0x05, 0xB8, 0xB3, 0x45, 0x06),
        array(0xD0, 0x2C, 0x1E, 0x8F, 0xCA, 0x3F, 0x0F, 0x02,
0xC1, 0xAF, 0xBD, 0x03, 0x01, 0x13, 0x8A, 0x6B),
        array(0x3A, 0x91, 0x11, 0x41, 0x4F, 0x67, 0xDC, 0xEA,
0x97, 0xF2, 0xCF, 0xCE, 0xF0, 0xB4, 0xE6, 0x73),
        array(0x96, 0xAC, 0x74, 0x22, 0xE7, 0xAD, 0x35, 0x85,
0xE2, 0xF9, 0x37, 0xE8, 0x1C, 0x75, 0xDF, 0x6E),
        array(0x47, 0xF1, 0x1A, 0x71, 0x1D, 0x29, 0xC5, 0x89,
0x6F, 0xB7, 0x62, 0x0E, 0xAA, 0x18, 0xBE, 0x1B),
        array(0xFC, 0x56, 0x3E, 0x4B, 0xC6, 0xD2, 0x79, 0x20,
0x9A, 0xDB, 0xC0, 0xFE, 0x78, 0xCD, 0x5A, 0xF4),
        array(0x1F, 0xDD, 0xA8, 0x33, 0x88, 0x07, 0xC7, 0x31,
0xB1, 0x12, 0x10, 0x59, 0x27, 0x80, 0xEC, 0x5F),
        array(0x60, 0x51, 0x7F, 0xA9, 0x19, 0xB5, 0x4A, 0x0D,
0x2D, 0xE5, 0x7A, 0x9F, 0x93, 0xC9, 0x9C, 0xEF),
        array(0xA0, 0xE0, 0x3B, 0x4D, 0xAE, 0x2A, 0xF5, 0xB0,
0xC8, 0xEB, 0xBB, 0x3C, 0x83, 0x53, 0x99, 0x61),
        array(0x17, 0x2B, 0x04, 0x7E, 0xBA, 0x77, 0xD6, 0x26,
0xE1, 0x69, 0x14, 0x63, 0x55, 0x21, 0x0C, 0x7D)
    );

```

Dalam melakukan proses invers sub byte untuk algoritma AES ditunjukkan pada coding berikut ini.

```

function InvSubByte($data){
    $state = $data;
    for($i=0; $i<$this->Nb; $i++){
        for($j=0; $j<4; $j++) {
            $state[$i][$j]=$this->invSub($state[$i][$j]);
        }
    }
}

```

```

    }
}
return $state;
}

```

#### c. *InvShiftRow*

Pada tahap ini data yang dihasilkan pada tahap sebelumnya (*SubBytes*) digeser ke kiri sesuai dengan coding di bawah ini.

```

function ShiftRow($data) {
    $state = array();
    for($i=0; $i<4; $i++){
        for($j=0; $j<$this->Nb; $j++) {
            $state[$i][$j] = $data[$i][($j+$i)%4];
        }
    }
    return $state;
}

```

#### d. *MixColumns*

Fungsi selanjutnya adalah fungsi mixcolumns, dalam fungsi ini terdapat empat kriteria yang menjadi syarat, yaitu:

- Jika *msb* data berawalan dengan 1 maka di *left shift* dan *xor* tetap
- Jika *msb* data berawalan dengan 0 maka di *left shift* tapi *xor* dihilangkan.
- Tetap untuk pengali 02 yaitu disubstitusi menjadi “*xor 1B*”
- Karena 03 (11) merupakan 10 *xor* 01 maka akan dipecah menjadi pengali 02 (yang berarti juga akan disubstitusi menjadi *xor 1B*) dan 01

Berikut ini adalah coding dalam melakukan mixcolumns pada algoritma AES.

```

function MixColumns($data) {
    $state = array();
    for ($i=0; $i < $this->Nb; $i++) {
        $state[0][$i] = $this->mult(0x02,$data[0][$i]) ^

```

```

$this->mult(0x03,$data[1][$i]) ^ $this->mult(0x01,$data[2][$i]) ^
$this->mult(0x01,$data[3][$i]);
    $state[1][$i] = $this->mult(0x01,$data[0][$i]) ^
$this->mult(0x02,$data[1][$i]) ^ $this->mult(0x03,$data[2][$i]) ^
$this->mult(0x01,$data[3][$i]);
    $state[2][$i] = $this->mult(0x01,$data[0][$i]) ^
$this->mult(0x01,$data[1][$i]) ^ $this->mult(0x02,$data[2][$i]) ^
$this->mult(0x03,$data[3][$i]);
    $state[3][$i] = $this->mult(0x03,$data[0][$i]) ^
$this->mult(0x01,$data[1][$i]) ^ $this->mult(0x01,$data[2][$i]) ^
$this->mult(0x02,$data[3][$i]);
    }
    return $state;
}

```

#### 4.1.4.2 Proses Kompresi dengan LZW

Setelah berhasil dienkrpsi maka hasil output binary nya akan dikompres menggunakan LZW.

```

function lzw_compress($string) {
    // compression
    $dictionary = array_flip(range("\0", "\xFF"));
    $word = "";
    $codes = array();
    for ($i=0; $i <= strlen($string); $i++) {
        $x = substr($string, $i, 1);
        if (strlen($x) && isset($dictionary[$word . $x])) {
            $word .= $x;
        } elseif ($i) {
            $codes[] = $dictionary[$word];
            $dictionary[$word . $x] = count($dictionary);
            $word = $x;
        }
    }

    // convert codes to binary string
    $dictionary_count = 256;
    $bits = 8; // ceil(log($dictionary_count, 2))
    $return = "";
    $rest = 0;
    $rest_length = 0;
    foreach ($codes as $code) {
        $rest = ($rest << $bits) + $code;
        $rest_length += $bits;
        $dictionary_count++;
        if ($dictionary_count >> $bits) {
            $bits++;
        }
    }

```



UNIVERSITAS ISLAM NEGERI

```

    while ($rest_length > 7) {
        $rest_length -= 8;
        $return .= chr($rest >>
            $rest_length);
        $rest &= (1 << $rest_length) - 1;
    }
    return $return . ($rest_length ? chr($rest <<
        (8 - $rest_length)) : "");
}

```

#### 4.1.4.3 Proses Dekompresi dengan LZW

Untuk melakukan dekripsi pertama-tama file akan dikonversi dulu ke binary kemudian dilakukan proses dekompresi dengan algoritma LZW.

```

function lzw_decompress($binary) {
    // convert binary string to codes
    $dictionary_count = 256;
    $bits = 8; // ceil(log($dictionary_count, 2))
    $codes = array();
    $rest = 0;
    $rest_length = 0;
    for ($i=0; $i < strlen($binary); $i++) {
        $rest = ($rest << 8) + ord($binary[$i]);
        $rest_length += 8;
        if ($rest_length >= $bits) {
            $rest_length -= $bits;
            $codes[] = $rest >> $rest_length;
            $rest &= (1 << $rest_length) - 1;
            $dictionary_count++;
            if ($dictionary_count >> $bits) {
                $bits++;
            }
        }
    }

    // decompression
    $dictionary = range("\0", "\xFF");
    $return = "";
    foreach ($codes as $i => $code) {
        $element = $dictionary[$code];
        if (!isset($element)) {
            $element = $word . $word[0];
        }
        $return .= $element;
        if ($i) {
            $dictionary[] = $word . $element[0];
        }
        $word = $element;
    }
    return $return;
}

```

#### 4.1.4.4 Proses Dekripsi dengan AES

Untuk proses deskripsi pada algoritma AES sendiri secara coding diimplementasikan sebagai berikut.

```

function decrypt($input){
    $data = str_split($input);
    $state = array();
    $count = 0;
    $this->pos_w = (($this->Nr+1)*4);
    $modulo = 0;

```

```

for($i=0; $i<4; $i++){
    for($j=0; $j<4; $j++) {
        $this->state[$i][$j] = ord($data[$count]);
        $count++;
    }
}

// AddRoundKey #1
$this->pos_w = ($this->pos_w)-4;
for($i=0; $i<4; $i++){
    $x = 0;
    for($j= $this->pos_w; $j < (($this->Nr+1)*4); $j++) {

        $y = $x++;

        $this->state[$i][$y] = $this->state[$i][$y] ^
$this->w[$i][$j];
    }
}

for ($i=0; $i<$this->Nr-1; $i++) {

    $this->state = $this->InvShiftRow($this->state);

    $this->state = $this->InvSubByte($this->state);

    $this->pos_w = ($this->pos_w)- $this->Nb;
    $this->state = $this->AddRoundKey($this->state);

    $this->state = $this->InvMixColumns($this->state);
}

$this->state = $this->InvShiftRow($this->state);

$this->state = $this->InvSubByte($this->state);

$this->pos_w = ($this->pos_w)- $this->Nb;
$this->state = $this->AddRoundKey($this->state);

```

```
$plain = "";
foreach($this->state as $state) {
    foreach($state as $data) {
        $plain .= chr($data);
    }
}
return $plain;
}
```

5 MB 10 MB 15 MB 20 MB 25 MB 30 MB 35 MB 40 MB 45 MB 50 MB

55 MB 60 MB 65 MB 70 MB 75 MB 80 MB 85 MB 90 MB 95 MB 100 MB

--	--	--	--	--	--	--	--	--	--	--

Tabel 4.3 Table Test Case Equivalent Class Partitioning

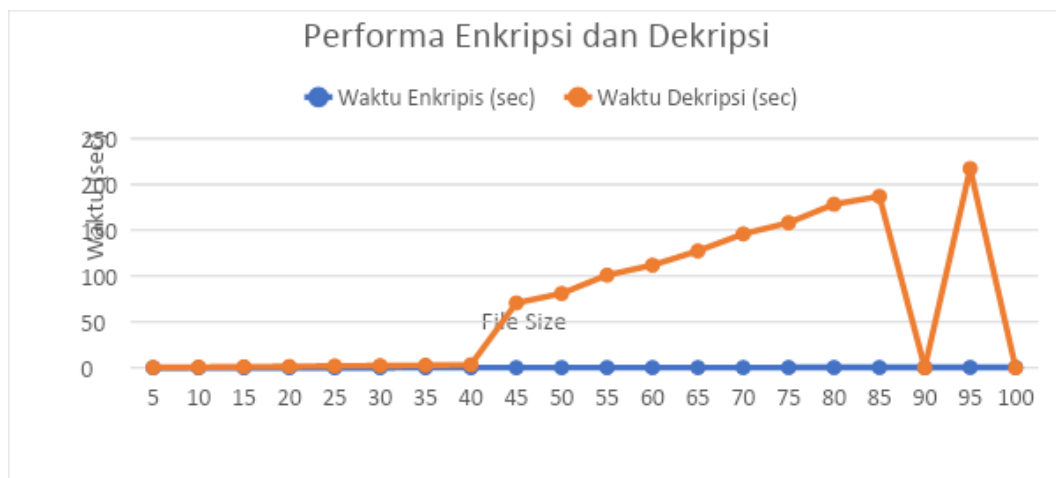
Test Case	Ukuran File	Partisi yang di tes	Waktu Enkripsi	Waktu Dekripsi	Keluaran
1	5 MB	ukuran file < 5 MB	0,0171 sec	0,3532 sec	Berhasil Upload
2	10 MB	5 MB >= ukuran file < 10 MB	0,0334 sec	0,6972 sec	Berhasil Upload
3	15 MB	ukuran file < 15 MB	0,0495 sec	1,0415 sec	Berhasil Upload
4	20 MB	15 MB >= ukuran file < 20 MB	0,0651 sec	1,4040 sec	Berhasil Upload
5	25 MB	ukuran file < 25 MB	0,1025 sec	1,9918 sec	Berhasil Upload
6	30 MB	25 MB >= ukuran file < 30 MB	0,1250 sec	2,3232 sec	Berhasil Upload
7	35 MB	ukuran file < 35 MB	0,1463 sec	2,7604 sec	Berhasil Upload
8	40 MB	35 MB >= ukuran file < 40 MB	0,1667 sec	3,2304 sec	Berhasil Upload
9	45 MB	ukuran file < 45 MB	0,1821 sec	70,6772 sec	Berhasil Upload
10	50 MB	45 MB >= ukuran file < 50 MB	0,2049 sec	81,0777 sec	Berhasil Upload
11	55 MB	ukuran file < 55 MB	0,2331 sec	101,0423 sec	Berhasil Upload
12	60 MB	55 MB >= ukuran file < 60 MB	0,2472 sec	111,9067 sec	Berhasil Upload
13	65 MB	ukuran file < 65 MB	0,2599 sec	127,4202 sec	Berhasil Upload
14	70 MB	65 MB >= ukuran file < 70 MB	0,2761 sec	146,0176 sec	Berhasil Upload
15	75 MB	ukuran file < 75 MB	0,3002 sec	158,1429 sec	Berhasil Upload
16	80 MB	75 MB >= ukuran file < 80 MB	0,3272 sec	178,3587 sec	Berhasil Upload
17	85 MB	ukuran file < 85 MB	0,3502 sec	186,8799 sec	Berhasil Upload
18	90 MB	85 MB >= ukuran file < 90 MB	0,3644 sec	Out of Memory	Gagal Upload
19	95 MB	ukuran file < 95 MB	0,3890 sec	217,2526 sec	Berhasil Upload



20	100 MB	95 MB >= ukuran file < 100 MB	0,4053 sec	Out of Memory	Gagal Upload
----	--------	-------------------------------	------------	---------------	--------------

Dari table test case diatas dapat dilihat bahwa terdapat keluaran berhasil dan keluaran gagal. Diantara keluaran yang berhasil dikarenakan waktu enkripsi rata-rata lebih kecil dibanding waktu dekripsi, sedangkan diantara keluaran yang gagal dikarenakan waktu yang diperlukan untuk dekripsi rata-rata lebih besar dan bisa juga disebabkan oleh jaringan internet yang lambat.

#### b. Perhitungan Akurasi Algoritma AES



Gambar 4.9 Performa Enkripsi dan Dekripsi

Dari grafik diagram garis diatas, hasil performa file yang akan di uji dengan ukuran yang berbeda-beda mempengaruhi *running time* proses enkripsi dan dekripsi. Kecepatan jaringan internet dapat mempengaruhi hasil *running time* proses. Berdasarkan diagram garis tersebut, proses dekripsi membutuhkan waktu lebih lama di bandingkan waktu proses enkripsi, hal ini dikarenakan untuk proses dekripsi semakin besar *file size* yang di *download* maka akan semakin lama waktu yang dibutuhkan. Pada diagram tersebut dapat dilihat bahwa pengujian algoritma ini dibatasi dengan *file size* 100 *Megabytes*, hal ini dikarenakan sistem ini belum mampu diuji oleh data sebanyak 100 *Megabytes* lebih.

Dari hasil pengujian algoritma diatas diperoleh perhitungan akurasi sebagai berikut:

$$Akurasi = \frac{Jumlah\ Total\ Berhasil}{Jumlah\ total\ berhasil + gagal} \times 100\%$$

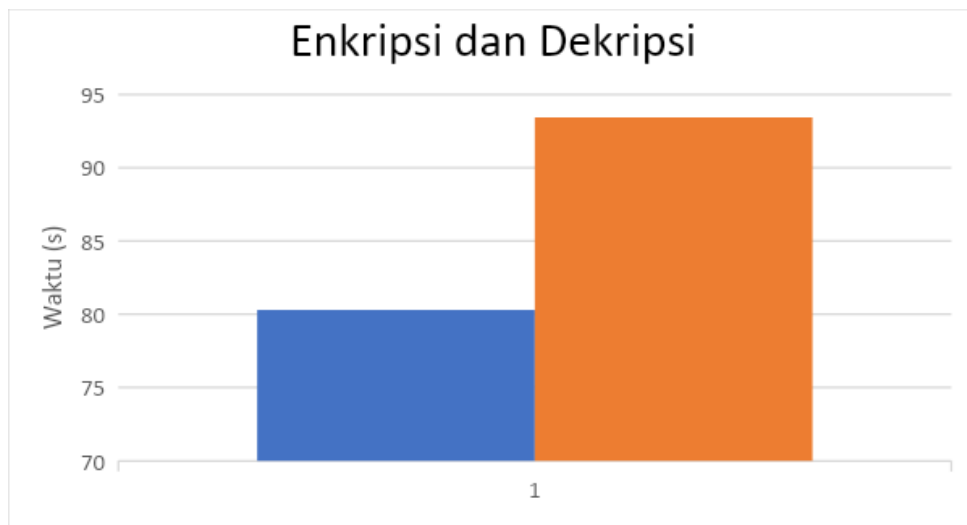
Rumus 4. 1 Perhitungan Akurasi Algoritma AES

Pada rumus diatas jumlah total berhasil adalah jumlah total pengujian dari metode enkripsi, dekripsi, kompresi dan dekompresi semuanya berhasil, apabila ada salah satu yang gagal maka pengujian dinyatakan gagal.

Berikut ini merupakan perhitungan tingkat akurasi sistem ini :

$$= \frac{18}{20} \times 100\% = 90\%$$

Jadi dapat disimpulkan bahwa tingkat akurasi metode AES-128 pada pengamanan *file Cloud Storage* adalah sebesar 90%.



Gambar 4.10 Waktu enkripsi dan dekripsi

Pada gambar 4.10 diatas menjelaskan waktu antara proses enkripsi dan dekripsi pada sistem ini dengan nilai rata-rata waktu proses enkripsi bernilai 80,3 detik (43%) dan rata-rata waktu proses dekripsi bernilai 93,43 detik (50%).

## 2. Pengujian performa algoritma LZW untuk compress dan decompress

### a. Equivalent Class Partitioning

Pada skenario pengujian ini data masukan berupa file dengan ukuran file dan ekstensi yang berbeda-beda.

5 MB	10 MB	15 MB	20 MB	25 MB	30 ME	35 MB	40 MB	45 MB	50 MB
55 MB	60 MB	65 MB	70 MB	75 MB	80 MB	85 MB	90 MB	95 MB	100 MB

--	--	--	--	--	--	--	--	--	--	--

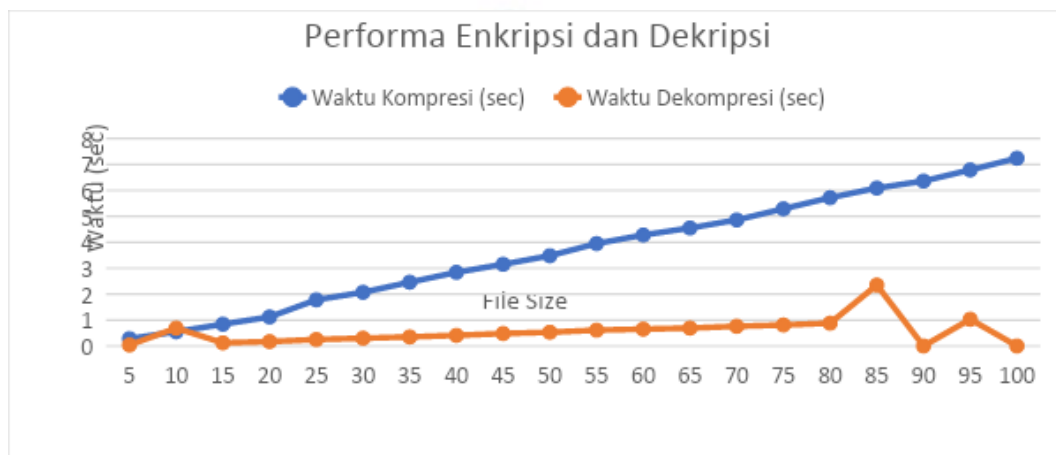
Tabel 4.4 Table Test Case Equivalent Class Partitioning

Test Case	Ukuran File	Partisi yang di tes	Waktu Kompresi	Waktu Dekompresi	Keluaran
1	5 MB	ukuran file < 5 MB	0,2876 sec	0,0424 sec	Berhasil Download
2	10 MB	5 MB >= ukuran file < 10 MB	0,5567 sec	0,6972 sec	Berhasil Download
3	15 MB	ukuran file < 15 MB	0,8438 sec	0,1270 sec	Berhasil Download
4	20 MB	15 MB >= ukuran file < 20 MB	1,1211 sec	0,1704 sec	Berhasil Download
5	25 MB	ukuran file < 25 MB	1,7894 sec	0,2568 sec	Berhasil Download
6	30 MB	25 MB >= ukuran file < 30 MB	2,0726 sec	0,3057 sec	Berhasil Download
7	35 MB	ukuran file < 35 MB	2,4656 sec	0,3573 sec	Berhasil Download
8	40 MB	35 MB >= ukuran file < 40 MB	2,8448 sec	0,4154 sec	Berhasil Download
9	45 MB	ukuran file < 45 MB	3,1536 sec	0,4860 sec	Berhasil Download
10	50 MB	45 MB >= ukuran file < 50 MB	3,4867 sec	0,5368 sec	Berhasil Download
11	55 MB	ukuran file < 55MB	3,9564 sec	0,6128 sec	Berhasil Download
12	60 MB	55 MB >= ukuran file < 60 MB	4,2829 sec	0,6505 sec	Berhasil Download
13	65 MB	ukuran file < 65 MB	4,5545 sec	0,6906 sec	Berhasil Download
14	70 MB	65 MB >= ukuran file < 70 MB	4,8645 sec	0,7608 sec	Berhasil Download
15	75 MB	ukuran file < 75 MB	5,2901 sec	0,8169 sec	Berhasil Download
16	80 MB	75 MB >= ukuran file < 80 MB	5,7223 sec	0,8814 sec	Berhasil Download
17	85 MB	ukuran file < 85 MB	6,0962 sec	2,3543 sec	Berhasil Download
18	90 MB	85 MB >= ukuran file < 90 MB	6,3641 sec	Out of Memory	Gagal Download

19	95 MB	ukuran file < 95 MB	6,7985 sec	1,0369 sec	Berhasil Download
20	100 MB	95 MB >= ukuran file < 100 MB	7,2481 sec	Out of Memory	Gagal Download

Dari table test case diatas dapat dilihat bahwa terdapat keluaran berhasil dan keluaran gagal. Diantara keluaran yang berhasil dikarenakan waktu dekompresi rata-rata lebih kecil dibanding waktu kompresi, sedangkan diantara keluaran yang gagal dikarenakan waktu yang diperlukan untuk dekompresi rata-rata lebih lama dan bisa juga disebabkan oleh jaringan internet yang lambat.

#### b. Perhitungan Akurasi Algoritma LZW



Gambar 4.11 Performa Kompresi dan Dekompresi

Dari grafik diagram garis diatas, hasil performa file yang akan di uji dengan ukuran yang berbeda-beda mempengaruhi *running time* proses kompresi dan dekompresi. Kecepatan jaringan internet dapat mempengaruhi hasil *running time* proses.

Pada diagram diatas dapat dilihat bahwa pengujian algoritma ini dibatasi dengan *file size* 100 *Megabytes*, hal ini dikarenakan sistem ini belum mampu diuji oleh data sebanyak 100 *Megabytes* lebih.

Dari hasil pengujian algoritma diatas diperoleh perhitungan akurasi sebagai berikut:

$$Akurasi = \frac{Jumlah\ Total\ Berhasil}{Jumlah\ total\ berhasil + gagal} \times 100\%$$

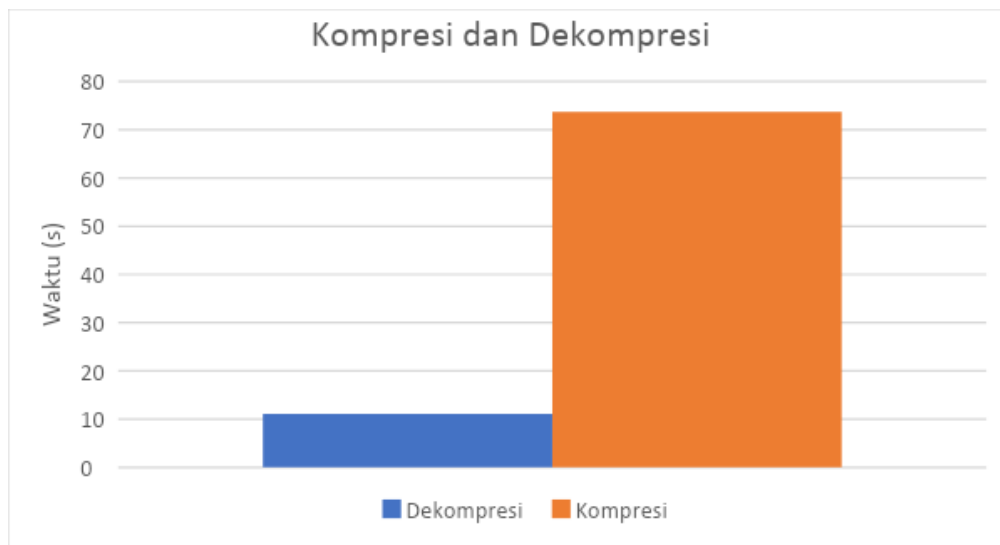
Rumus 4. 2 Perhitungan Akurasi Algoritma LZW

Pada rumus diatas jumlah total berhasil adalah jumlah total pengujian dari metode enkripsi, dekripsi, kompresi dan dekompresi semuanya berhasil, apabila ada salah satu yang gagal maka pengujian dinyatakan gagal.

Berikut ini merupakan perhitungan tingkat akurasi sistem ini :

$$= \frac{18}{20} \times 100\% = 90\%$$

Jadi dapat disimpulkan bahwa tingkat akurasi metode kompresi LZW pada pengamanan *file Cloud Storage* adalah sebesar 92%.



Gambar 4.12 Waktu kompresi dan dekompresi

Pada gambar 4.12 diatas menjelaskan waktu antara proses kompresi dan dekompresi pada sistem ini dengan nilai rata-rata waktu proses kompresi bernilai 73,7 detik (50%) dan rata-rata waktu proses dekompresi bernilain 11,12 detik (42%).

### 4.2.3 Rencana Pengujian

Rencana pengujian berguna sebagai landasan utama dalam melakukan pengujian sistem. Berikut ini adalah rencana pengujian sistem yang telah dibuat ditampilkan tabel 4.5 berikut.

Tabel 4.5 Rencana Pengujian Sistem

Kelas Pengujian	Butir Uji	Jenis Pengujian	Hasil Pengujian
Login	Inputan password	Black Box	Sukses
	Validasi inputan	Black Box	Sukses

	Verifikasi password	<i>Black Box</i>	<i>Sukses</i>
Registrasi	Inputan registrasi	<i>Black Box</i>	<i>Sukses</i>
	Validasi inputan	<i>Black Box</i>	<i>Sukses</i>
	Verifikasi email	<i>Black Box</i>	<i>Sukses</i>
List File Upload	View file upload	<i>Black Box</i>	<i>Sukses</i>
	Download	<i>Black Box</i>	<i>Sukses</i>
	Hapus data	<i>Black Box</i>	<i>Sukses</i>
Proses Upload data	Browse data dari PC	<i>Black Box</i>	<i>Sukses</i>
	Verifikasi password	<i>Black Box</i>	<i>Sukses</i>
	Proses unggah file	<i>Black Box</i>	<i>Sukses</i>

Hasil pengujian keseluruhan fungsi-fungsi yang terdapat pada sistem yang dibuat di tampilkan pada tabel 4.6 berikut.

Tabel 4.6 Pengujian Fungsionalitas

ID Testing	Testing Status	Page / Area	Test description	Step Testing	Expected Result	Test Result
L-1	Positif	Halaman Login	Masuk ke akun	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Klik Navigasi "Login" 3. Sistem menampilkan halaman login 4. Memasukan email dan password 4. Klik "Masuk"	Sistem akan melakukan autentifikasi  Jika valid, maka aktor akan dilempar ke halaman utama	TRUE
L-2	Negatif	Halaman Login	Jika email tidak terdaftar	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">book-https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Memasukan email dan password 4. Klik "Masuk"	Sistem menampilkan notifikasi, email tidak terdaftar	TRUE
L-3	Negatif	Halaman Login	Jika password salah	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Memasukan email dan password 4. Klik "Masuk"	Sistem akan menampilkan notifikasi, password tidak valid	TRUE
R-1	Positif	Halaman Registrasi	Melakukan proses registrasi	1. Aktor mengakses <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Aktor mengklik tombol "Register" 4. Sistem menampilkan halaman	1. Sistem melakukan pengecekan email, Jika sudah terdaftar maka	TRUE

				registrasi 5. Aktor memasukkan inputan registrasi dengan benar 7. Mengklik tombol "Daftar"	muncul notifikasi email sudah terdaftar	
--	--	--	--	--	---	--

Tabel 4.7 Pengujian Fungsionalitas (Lanjutan)

ID Testing	Testing Status	Page / Area	Test description	Step Testing	Expected Result	Test Result
					2. Jika berhasil, muncul notifikasi registrasi berhasil dan diarahkan ke halaman login  3. sistem mengirim notifikasi registrasi ke email	
R-2	Negatif	Halaman Registrasi	Registrasi dengan email yang sudah terdaftar	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Aktor mengklik tombol "Register" 4. Sistem menampilkan halaman registrasi 5. Aktor memasukkan inputan email yang sudah terdaftar sebelumnya 6. Mengklik tombol "Daftar"	Muncul notifikasi email sudah terdaftar dan proses registrasi dihentikan	TRUE
R-3	Negatif	Halaman Registrasi	validasi form registrasi jika data kosong	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Aktor mengklik tombol "Register"	1. Sistem menampilkan notifikasi data yang kosong. Pada setiap	TRUE

				4. Sistem menampilkan halaman registrasi 5. Aktor tidak memasukan data inputan registrasi 6. Mengklik tombol "Daftar"	komponen form akan berubah menjadi merah yang menunjukkan data tersebut "required"	
R-4	Negatif	Halaman Registrasi	validasi format email dan nomor telepon	1. Aktor mengakses URL <a href="https://pimenvibritania.tech/login">https://pimenvibritania.tech/login</a> 2. Sistem menampilkan halaman login 3. Aktor mengklik tombol "Register"	Muncul notifikasi untuk memasukan email yang benar	TRUE

Tabel 4.8 Pengujian Fungsionalitas (Lanjutan)

ID Testing	Testing Status	Page / Area	Test description	Step Testing	Expected Result	Test Result
				4. Sistem menampilkan halaman registrasi 5. Aktor memasukan email yang tidak sesuai format baku 6 Mengklik tombol "Daftar"		
F-1	Positif	List File Upload	View File upload	Aktor membuka halaman file upload	Sistem menampilkan file yang diupload oleh user	TRUE
F-2	Positif	List File Upload	View File upload	1. Aktor berada dihalaman list file upload 2. Aktor mengklik tombol "Download"	Sistem melakukan proses download file	TRUE
F-3	Positif	List File Upload	View File upload	1. Aktor berada dihalaman list file upload 2. Aktor mengklik tombol "Hapus"	File yang dipilih dihapus dari sistem	TRUE
U-1	Positif	Upload Data	Proses upload Data	1. Aktor mengklik menu upload file 2. Aktor mengklik tombol browse file 3. Mengisi password 4. Mengklik tombol upload	File diupload ke dalam cloud storage	TRUE
U-2	Negatif	Upload Data	Proses upload Data	1. Aktor mengklik menu upload file 2. Aktor tidak mengklik tombol browse file 3. Mengisi password 4. Mengklik tombol upload	Muncul notifikasi untuk memilih file yang akan diupload	TRUE
U-3	Negatif	Upload Data	Proses upload	1. Aktor mengklik menu upload file	Muncul notifikasi	TRUE



			Data	2. Aktor mengklik tombol browse file 3. Mengisi password yang salah 4. Mengklik tombol upload	password salah	
--	--	--	------	---	----------------	--

Tabel 4.6 sampai tabel 4.8 dilakukan pengujian fungsionalitas sistem yang dibuat. Pengujian ini bermaksud untuk mengecek apakah sistem berjalan dengan baik atau tidak. Dalam tabel tersebut terdapat kolom *test result* dengan nilai TRUE yang menandakan sistem berjalan dengan baik. Selain itu dalam tabel tersebut juga terdapat kolom *testing status* yang digunakan untuk mengetahui respon dari sistem ketika pengguna mengakses sistem tersebut, seperti nilai negatif untuk mengetahui respon sistem ketika pengguna melakukan kesalahan seperti memasukan password salah, email sudah terdaftar, email salah dan lain-lain, maka sistem akan memunculkan pesan error sesuai kesalahan pengguna. Sebaliknya nilai positif untuk mengetahui respon sistem ketika pengguna tidak melakukan kesalahan, maka sistem akan memunculkan pesan sukses.