BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi Sistem adalah tahapan membangun sistem sesuai dengan perancangan yang telah dilakukan sehingga dapat memenuhi tujuan pembangunan sebuah sistem, tahapan implementasi pada penelitian ini terbagi menjadi dua bagian yaitu implementasi fitur verifikasi perizinan berbasis QR code dan pengamanan sesi komunikasi dengan menggunakan TLS.

4.1.1 Implementasi Fitur Verifikasi Perizinan Berbasis QR Code

Berikut adalah penjelasan dari implementasi yang dilakukan untuk menambahkan fitur verifikasi perizinan berbasis QR code pada SIMPPTSP.

4.1.1.1 Implementasi Penambahan Tabel Basis Data

Implementasi perubahan basis data dilakukan untuk menambahkan tabel baru agar dapat diterapkan fitur verifikasi perizinan, penambahan tabel dilakukan pada basis data SIMPPTSP yang merupakan basis data MYSQL. Dilakukan implementasi dari hasil perancangan penambahan tabel basis data dengan menggunakan perintah sebagai berikut.

```
-- ----------------------------
-- Table structure for verifikasi
-- ----------------------------
DROP TABLE IF EXISTS `verifikasi`;
CREATE TABLE `verifikasi` (
  `id_pemohon` int(10) unsigned NOT NULL,
  `kode_akses` int(8) DEFAULT NULL,
  `nomor_warkah` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_pemohon`),
  CONSTRAINT `verifikasi_ibfk_1` FOREIGN KEY (`id_pemohon`)
REFERENCES `pemohon` (`id_pemohon`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Perintah yang dijalankan menghasilkan tabel baru dengan nama verifikasi, tabel ini digunakan untuk menyimpan nomor warkah (*username*) dan kode akses (*password*)

yang nantinya digunakan untuk verifikasi akses saat dilakukan pengakses data perizinan baik itu melalui layanan *web* ataupun halaman *web*.

4.1.1.2 Implementasi Penerapan QR Code

Implementasi penerapan QR code dilakukan dengan mengubah aktivitas pencetakan agar menyertakan QR code. Pada Tabel 4.1 dijelaskan perubahan yang dilakukan pada *class* SIMPPTSP yang diubah agar dapat menyertakan QR code perizinan.

Tabel 4.1 Implementasi perubahan pada class SIMPPTSP

Nama Class : SiupController	
Visibility Method dan Nama Method	**Deskripsi**
Public actionPencetakan(id)	Digunakan untuk memproses permintaan pencetakan perizinan SIUP
Nama Class : TdpController	
Visibility Method dan Nama Method	**Deskripsi**
Public actionPencetakan(id)	Digunakan untuk memproses permintaan pencetakan perizinan TDP
Nama class : Pengaturan	
Visibility Method dan Nama Method	**Deskripsi**
public getPrivateKeyBPMPPT()	Mengembalikan nilai dari *record private key* yang tersimpan pada tabel pengaturan, jika *private key* belum ada maka dilakukan pembangkitan pasangan kunci.

Sedangkan implementasi pembangkitan QR code yang diamankan dengan menggunakan *digital signature* di representasikan dengan menggunakan *class* – *class* yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Implementasi class penerapan QR code

Nama class : Encoder	
Visibility Method dan Nama Method	**Deskripsi**
Public Encoder(nomorResi, nomorWarkah,jenisIzin,privatekey)	Konstrukor dari *class*, digunakan untuk inisialisasi nilai dari class saat dilakukan instansiasi class, *method* ini dipanggil di dalam *method* actionPencetakan() baik itu oleh class SiupController ataupun TdpController.
Public setKodeAkses()	Digunakan untuk membangkitkan kode akses.
Private setPesan()	Digunakan untuk menyusun pesan yang terdiri dari nomor warkah dan kode akses.
Private setDigitalSignature()	Digunakan untuk membangkitkan *digital signature* dari pesan dengan hasil keluaran *signature* dalam format base64.
Private setQRCode()	Digunakan untuk membangkitkan QR code dengan hasil keluaran gambar dalam format base64.

Public getNomorWarkah()	Mengembalikan nilai dari *attribute* nomor warkah dari *instance class* encoder.
Public getKodeAkses()	Mengembalikan nilai dari *attribute* kode akses dari *instance class* encoder.
Public getQRCode()	Mengembalikan nilai dari *attribute* qrCode dari *instance class* encoder.

4.1.1.3 Implementasi Layanan Web

Implementasi dari layanan *web* direpresentasikan dengan menggunakan class yang dijelaskan pada Tabel 4.3.

Tabel 4.3 Implementasi class layanan web

Nama class : VerifikasiController	
Visibility Method dan Nama Method	**Deskripsi**
Public actionApi()	Digunakan untuk menerima permintaan data perizinan dan mengirimkan respon sesuai dengan hasil pemrosesan.
Nama class : Verifikasi	
Visibility Method dan Nama Method	**Deskripsi**
Public validasiAkses(nomorWarkah, kodeAkses)	Digunakan untuk memeriksa validitas kombinasi dari nomor warkah dan kode akses
Public getDataPerizinanByNomorWarkah(nomorWarkah)	Mengembalikan data perizinan dari suatu perizinan berdasarkan nomor warkah.

4.1.1.4 Implementasi Verifikasi

Implementasi verifikasi terbagi menjadi dua bagian yaitu implementasi verifikasi perizinan melalui aplikasi *mobile* dan melalui halaman *web*, dengan penjelasan sebagai berikut:

1. Implementasi aplikasi *mobile* verifikasi perizinan

   Implementasi dari aplikasi *mobile* verifikasi perizinan dijelaskan pada Tabel 4.4.

   Tabel 4.4 Implementasi aplikasi mobile verifikasi perizinan

Nama Object : Decoder.view	
onClick()	Digunakan untuk menerima *event click* pada tombol pindai QR code.
Nama Object : Decoder.controller	
Nama function	**Deksripsi**
scanBarcode()	Digunakan untuk memindai *barcode*, pemindaian dilakukan dengan menggunakan kamera dan bantuan *library cordova barcode scanner*.
preProcess(dataQRCode)	Digunakan untuk memisahkan data hasil pembacaan QR code menjadi , *signature*,

	nomor warkah dan kode akses. Untuk kemudian dilakukan validasi struktur data
verifiyDigitalSignature(data)	Digunakan untuk melakukan validasi digital *signature* dari QR code, validasi dilakukan dengan bantuan *library jsrsasign* dan *public key* BPMPPT.
transform(key)	Digunakan untuk mengubah *key json* hasil dari *web service* menjadi label.
Nama Object : Decoder.provider	
Nama function	**Deskripsi**
getHeader(data)	Digunakan untuk menambahkan *header basic authentication* pada permintaan verifikasi perizinan yang akan dikirimkan ke layanan *web*.
Nama Object : Decoder.service	
Nama function	**Deskripsi**
fetch(data)	Digunakan mengirimkan permintaan verifikasi perizinan ke layanan *web*.

2. Implementasi halaman web verifikasi perizinan

   Implementasi dari halaman *web* verifikasi perizinan direpresentasikan dengan menggunakan *class-class* yang dijelaskan pada Tabel 4.5.

Tabel 4.5 Implementasi class halaman web verifikasi

Nama class : VerifikasiController	
Visibility Method dan Nama Method	**Deskripsi**
Public actionIndex()	Digunakan untuk menampilkan formulir verifikasi perizinan dan menampilkan data perizinan.
Public actionError()	Digunakan untuk menampilkan pesan kesalahan jika nomor warkah atau kode akses yang masukan tidak valid sehingga data perizinan tidak ditemukan.
Nama class : Verifikasi	
Visibility Method dan Nama Method	**Deskripsi**
Public validasiAkses(nomorWarkah, kodeAkses)	Digunakan untuk memeriksa validitas kombinasi dari nomor warkah dan kode akses
Public getDataPerizinanByNomorWarkah(nomorWarkah)	Mengembalikan data perizinan dari suatu perizinan berdasarkan nomor warkah.

4.1.1.5 Implementasi Antar Muka

Implementasi antar muka terbagi menjadi dua bagian yaitu implementasi antar muka aplikasi *mobile* verifikasi perizinan dan implementasi antar muka halaman *web* verifikasi, dengan penjelasan sebagai berikut:

1. Implementasi antar muka aplikasi *mobile* verifikasi perizinan

   Antar muka aplikasi *mobile* verifikasi perizinan meliputi halaman utama aplikasi yang memuat tombol yang digunakan untuk melakukan pemindaian QR code dan tabel yang digunakan untuk menampilkan data perizinan yang diterima dari layanan *web*. Pada Tabel 4.6 dijelaskan implementasi dari antarmuka aplikasi *mobile* verifikasi perizinan.

   Tabel 4.6 Implementasi antar muka aplikasi mobile verifikasi perizinan

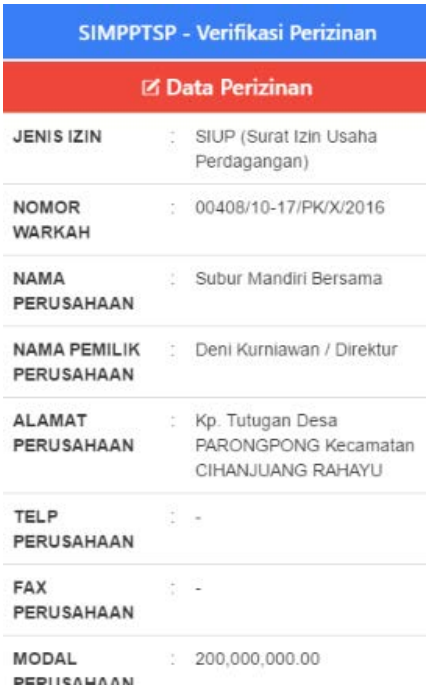
Nama Antar Muka	Nama File	Deskripsi
Halaman utama	Index.html	Digunakan untuk menampilkan tombol yang digunakan untuk memindai QR code dan tabel yang digunakan untuk menampilkan data perizinan yang diterima dari layanan *web*.

   Tampilan dari halaman utama dari aplikasi *mobile* verifikasi perizinan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi halaman utama aplikasi mobile verifikasi perizinan

   Sedangkan untuk tampilan yang menyertakan tabel berisi data perizinan dapat dilihat pada Gambar 4.2.



Gambar 4.2 Implementasi halaman utama (tampilan data perizinan) aplikasi mobile verifikasi perizinan

2. Implementasi antar muka halaman web verifikasi perizinan

   Antar muka halaman *web* verifikasi perizinan meliputi formulir untuk verifikasi perizinan, tampilan data perizinan dan tampilan pesan kesalahan jika data perizinan tidak temukan. Pada Tabel 4.7 dijelaskan implementasi dari antarmuka halaman *web* verifikasi perizinan.

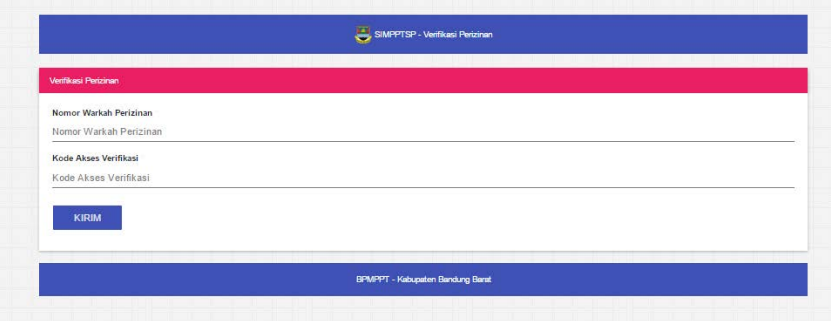
   Tabel 4.7 Implementasi antar muka halaman web verifikasi perizinan

Nama Antar Muka	Nama File	Deskripsi
Formulir verifikasi perizinan dan Tampilan Data perizinan	index.php	Digunakan untuk menampilkan formulir verifikasi perizinan dan menampilkan data perizinan jika verifikasi perizinan berhasil dilakukan
Pesan kesalahan	error.php	Digunakan untuk menampilkan pesan kesalahan jika verifikasi perizinan gagal dilakukan

   Dengan tampilan antar muka sebagai berikut:

   a. Formulir verifikasi perizinan

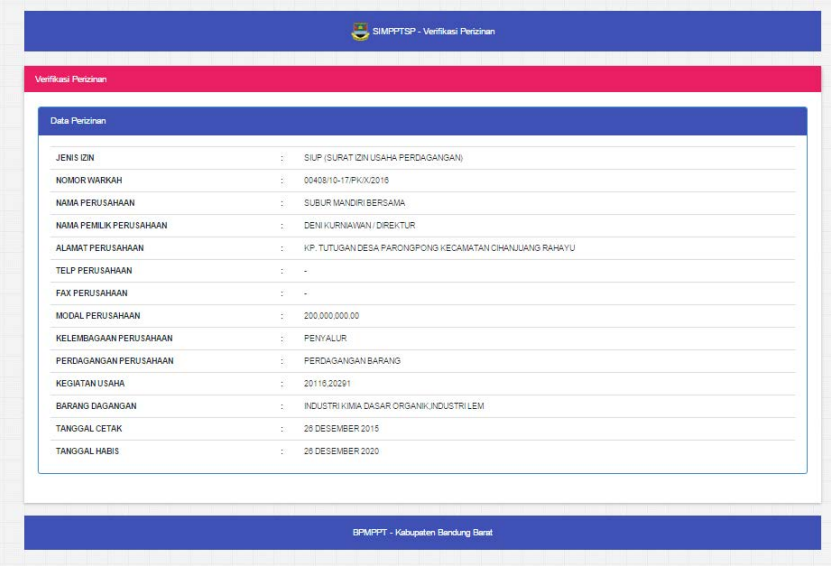
      Tampilan dari formulir verifikasi perizinan dapat dilihat pada Gambar 4.3



Gambar 4.3 Implementasi antar muka formulir verifikasi perizinan

b. Tampilan data perizinan

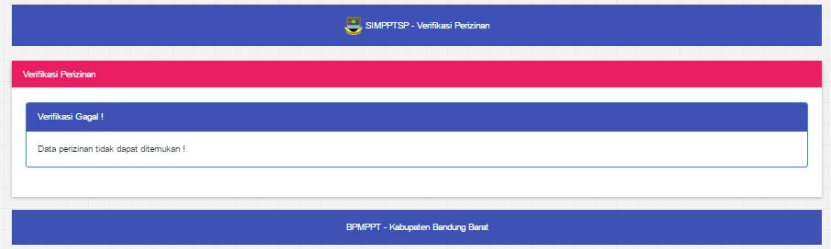
Tampilan dari data perizinan dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi antar muka tampilan data perizinan

c. Pesan kesalahan

Tampilan dari pesan kesalahan dapat dilihat pada Gambar 4.5.



Gambar 4.5 Implementasi antar muka pesan kesalahan

4.1.2 Implementasi Pengamanan Dengan TLS

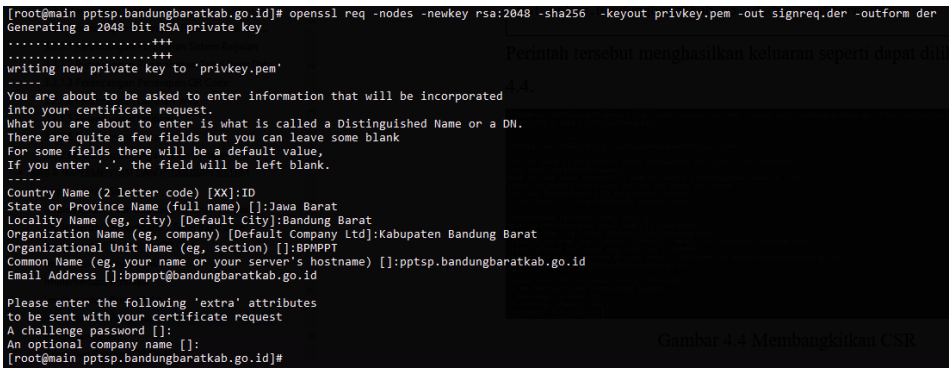
Implementasi pengamanan dengan TLS dilakukan dengan melakukan perubahan konfigurasi pada *web server* yang digunakan untuk menjalankan SIMPPTSP sehingga dapat berjalan di atas protokol *https* (*hypertext transfer protocol secure*) dengan implementasi sebagai berikut:

1. Membangkitkan CSR

   Pembangkitan CSR dilakukan dengan menggunakan perintah sebagai berikut

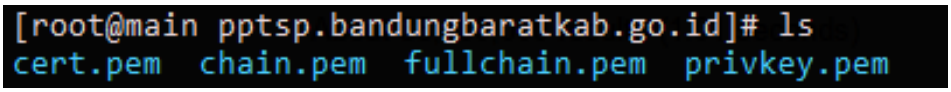
   ```
   openssl req -nodes -newkey rsa:2048 –Sha256  -keyout privkey.pem -
   out pptsp.bandungbaratkab.go.id.der –outform der
   ```

   Perintah tersebut menghasilkan keluaran seperti dapat dilihat pada Gambar 4.6.



Gambar 4.6 Membangkitkan CSR

   Hasil dari perintah adalah *file* CSR dan *private key*, *file* CSR dalam format DER , *file* CSR kemudian diserahkan kepada CA untuk ditandatangani yang menghasilkan *file* sertifikat SSL beserta *file chain certificate* yang dapat dilihat pada Gambar 4.7.



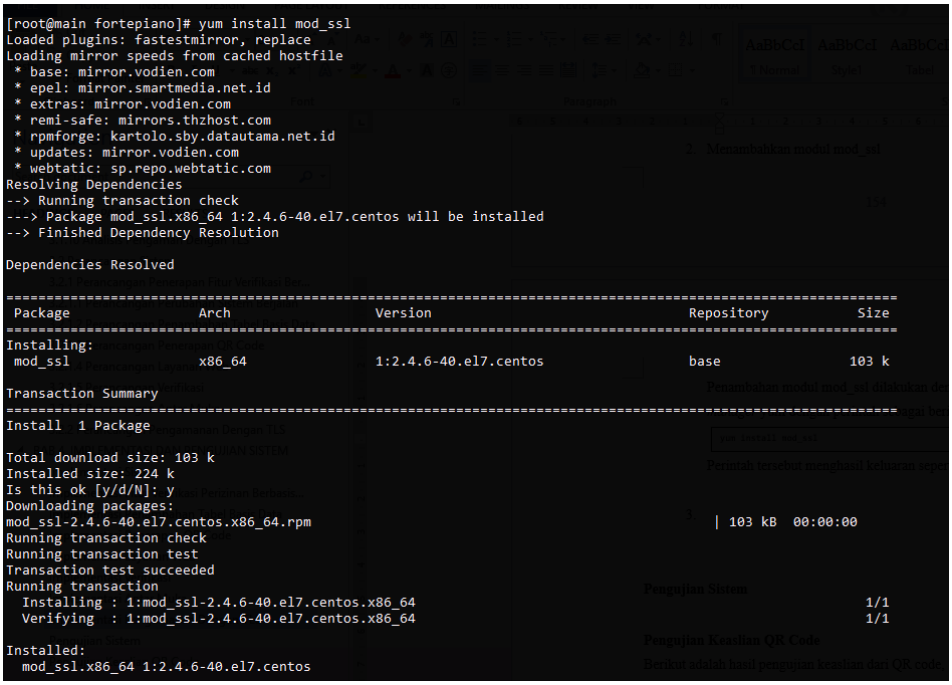
Gambar 4.7 Sertifikat SSL beserta kelengkapanya

2. Menambahkan modul mod_ssl

   Penambahan modul mod_ssl dilakukan dengan menggunakan *package manager* yum dengan perintah sebagai berikut:

   ```
   yum install mod_ssl
   ```

Perintah tersebut menghasil keluaran seperti dapat dilihat pada Gambar 4.8.



Gambar 4.8 Instalasi mod_ssl

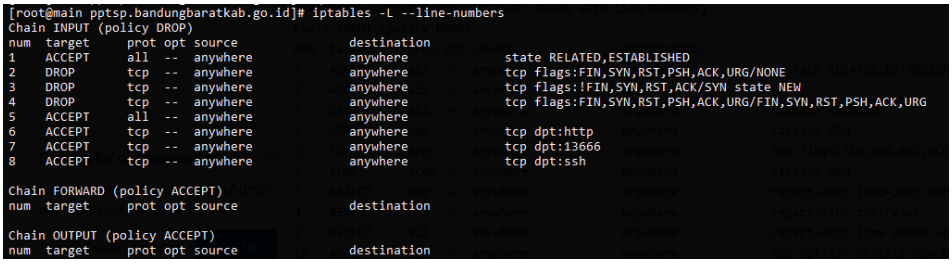
Paska instalasi, *mod_ssl* secara otomatis langsung diaktifkan dan siap digunakan.

3. Membuka *Port 443* pada *firewall*

Perangkat lunak *firewall* yang digunakan oleh *server* adalah *iptables*, sebelum menambahkan aturan baru pada *firewall*, ditampilkan konfigurasi yang berjalan yang dapat dilakukan dengan perintah sebagai berikut:

```
iptables -L -line-numbers
```

Perintah tersebut menghasilkan keluaran seperti dapat dilihat pada Gambar 4.9.

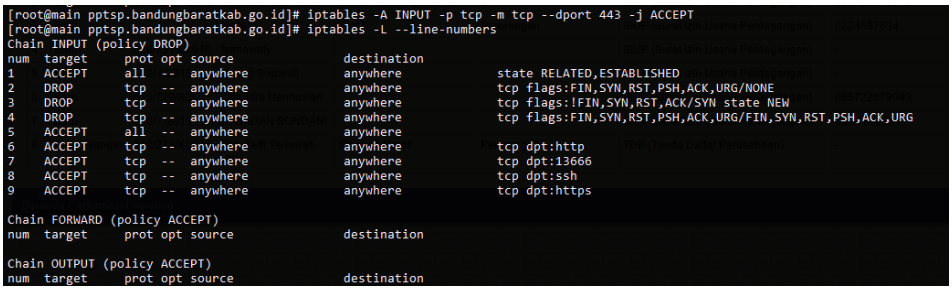


Gambar 4.9 Pengaturan firewall berjalan

Dilakukan penambahan aturan baru agar *server* membuka *port* 443(*https*), penambahan aturan dilakukan dengan menggunakan perintah sebagai berikut:

```
iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
iptables –L –line-numbers
```

Perintah tersebut menghasilkan keluaran seperti dapat dilihat pada Gambar 4.10.

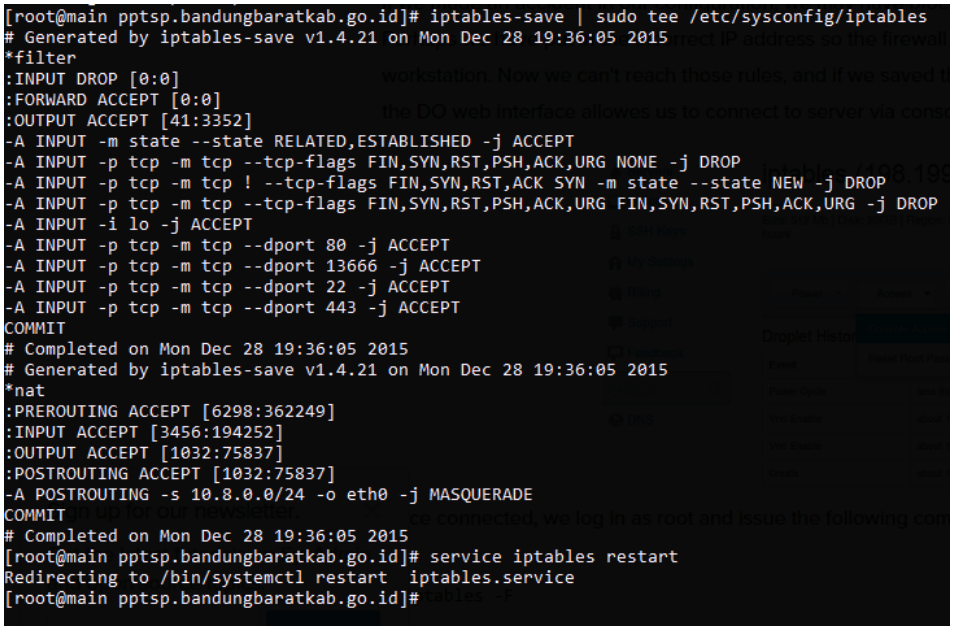


Gambar 4.10 Membuka port 443 pada firewall

Seperti terlihat pada Gambar 4.10 aturan baru ditambahkan pada *firewall* sehingga *firewall* membuka *port* 443(*https*), kemudian dilakukan penyimpanan konfigurasi dan *restart service firewall* dengan menggunakan perintah sebagai berikut:

```
iptables-save | sudo tee /etc/sysconfig/iptables
service iptables restart
```

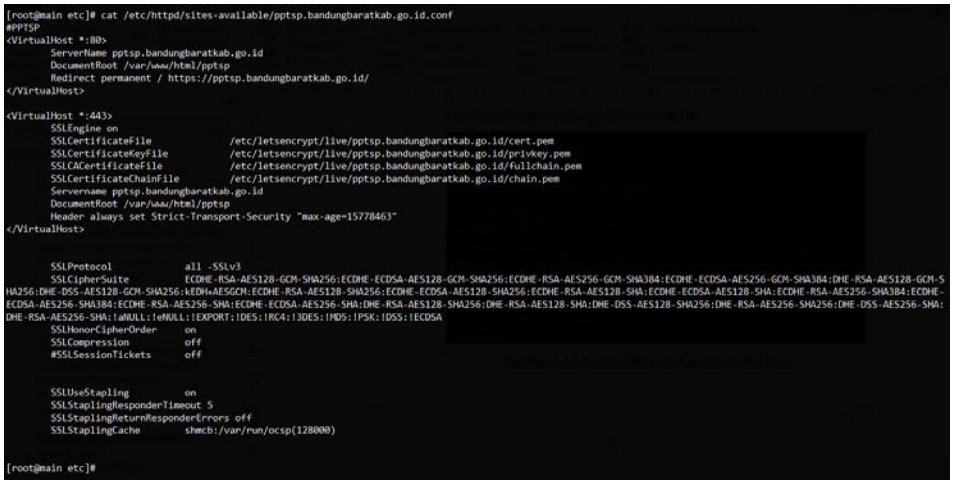
perintah tersebut menghasilkan keluaran seperti dapat dilihat pada Gambar 4.11.



Gambar 4.11 Menyimpan konfigurasi dan restart service firewall

4. Menambahkan konfigurasi virtual host dan mengalihkan permintaan dari Port 80 ke Port 443.

   Dilakukan perubahan pada konfigurasi *virtual host*, *webserver* yang digunakan adalah apache 2.4.6, untuk menambahkan konfigurasi *virtual host* dengan menambahkan *file /etc/httpd/sites-enable/pptsp.bandungbaratkab.go.id.conf* , penambahan dilakukan sesuai dengan rancangan yang telah dibahas pada sub bab perancangan pengamanan dengan TLS , berikut hasil dari penambahan konfigurasi *virtual host* dapat dilihat pada Gambar 4.12.

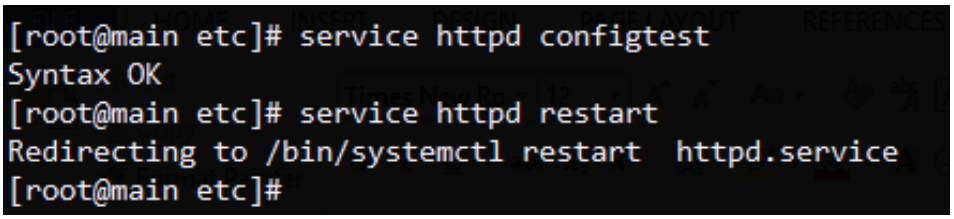


Gambar 4.12 Penambahan konfigurasi virtual host

Kemudian konfigurasi disimpan dan dilakukan *restart service httpd* dengan menggunakan perintah sebagai berikut:

```
Service httpd configtest
Service httpd restart
```

Perintah tersebut menghasilkan keluaran seperti dapat dilihat pada Gambar 4.13.



Gambar 4.13 Restart service httpd

4.2 Pengujian Sistem

Pengujian sistem adalah tahapan di mana dilakukan pengujian fungsionalitas terhadap hasil implementasi pada tahapan pengujian juga ditentukan apakah sistem sudah dapat memenuhi tujuan pembangunan sistem tersebut.

4.2.1 Rencana Pengujian Fitur Verifikasi Perizinan Berbasis QR code

Pada tabel Tabel 4.8 dijelaskan rincian dari rencana pengujiaan fitur verifikasi perizinan berbasis QR code.

Tabel 4.8 Rencana pengujian fitur verifikasi perizinan berbasis QR code

Pengujian	Butir Uji	Jenis Pengujian
Penerapan QR Code	Mencetak perizinan yang disertai QR code perizinan untuk perizinan jenis SIUP.	Blackbox
	Mencetak perizinan yang disertai QR code perizinan untuk Perizinan jenis TDP.	Blackbox
Layanan Web	Mengirimkan permintaan data perizinan tanpa nomor warkah dan kode akses	Blackbox
	Mengirimkan permintaan data perizinan dengan nomor warkah dan kode akses yang tidak valid	Blackbox
	Mengirimkan permintaan data perizinan dengan nomor warkah dan kode akses valid	Blackbox
Verifikasi Perizinan melalui halaman *web* verifikasi perizinan	Memasukan nomor warkah atau / dan kode akses yang tidak valid	Blackbox
	Memasukan nomor warkah dan kode akses valid	Blackbox
Verifikasi Perizinan melalui aplikasi *mobile* verifikasi perizinan	Memindai QR code perizinan dengan struktur tidak valid	Blackbox
	Memindai QR code perizinan dengan signature tidak valid	Blackbox
	Memindai QR code perizinan dengan data perizinan tidak ditemukan	Blackbox
	Memindai QR code perizinan valid	Blackbox
Pengujian kompatibilitas aplikasi *mobile* verifikasi perizinan	Pengujian installasi aplikasi *mobile* verifikasi perizinan dengan menggunakan emulator android.	Blackbox

4.2.2 Pengujian Fitur Verifikasi Perizinan Berbasis QR Code

Pengujian yang dilakukan adalah menguji hasil pengujian implementasi fitur verifikasi perizinan berbasis QR code yang telah diterapkan, dengan hasil sebagai berikut:

1. Pengujian penerapan QR code

   Hasil dari pengujian penerapan QR code dijelaskan pada Tabel 4.9.

Tabel 4.9 Hasil pengujian penerapan QR code

Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Perintah cetak perizinan jenis SIUP	Sistem mengirimkan *file* perizinan SIUP yang disertai QR code	Sistem mengirimkan *file* perizinan yang disertai QR code	Diterima
Perintah cetak perizinan jenis TDP	Sistem mengirimkan *file* perizinan TDP yang disertai QR code	Sistem mengirimkan *file* perizinan TDP yang disertai QR code	Diterima

2. Pengujian layanan web

Hasil dari pengujian penerapan layanan *web* dijelaskan pada Tabel 4.10

Tabel 4.10 Hasil pengujian layanan web

Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Permintaan dengan nomor warkah dan kode akses kosong. Nomor warkah : *Null*, Kode akses *Null*	Layanan *web* mengirimkan *http status code* 401, yang artinya layanan menolak memproses permintaan dikarenakan tidak ada data otorisasi.	*Client* menerima *http status code* 401, yang artinya layanan menolak memproses permintaan dikarenakan tidak ada data otorisasi.	Diterima
Permintaan dengan nomor warkah dan/atau kode akses tidak valid. Nomor warkah : "1234-2525-2525" Kode akses : "12345678"	Layanan *web* mengirimkan *http status code 403*, yang artinya layanan menerima permintaan namun menolak memenuhi.	*Client* menerima http status code 403, yang artinya layanan menerima permintaan namun menolak memenuhi.	Diterima
Permintaan dengan nomor warkah dan kode akses valid. Nomor warkah : "00408/10-17/PK/X/2015: Kode akses : "51029408"	Layanan *web* mengirimkan *http status code 200* yang artinya layanan menerima permintaan dan mengirimkan data balasan berupa data perizinan yang bersangkutan	*Client* menerima data perizinan yang bersangkutan	Diterima

3. Pengujian verifikasi perizinan melalui halaman web verifikasi perizinan

Hasil dari pengujian penerapan verifikasi perizinan melalui halaman *web* dijelaskan pada Tabel 4.11.

Tabel 4.11 Hasil pengujian verifikasi perizinan melalui halaman web verifikasi perizinan

Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
Nomor warkah dan/atau kode akses tidak valid. Nomor warkah : "1234-2525-2525"	Menampilkan halaman kesalahan data perizinan tidak ditemukan	Menampilkan halaman kesalahan data perizinan tidak ditemukan	Diterima

Kode akses : “12345678”			
Nnomor warkah dan kode akses valid. Nomor warkah : “00408/10-17/PK/X/2015: Kode akses : “51029408”	Menampilkan data perizinan yang bersangkutan	Menampilkan data perizinan yang bersangkutan	Diterima

4. Pengujian verifikasi perizinan melalui aplikasi *mobile verifikasi perizinan*

   Verifikasi perizinan melalui aplikasi *mobile* verifikasi perizinan dilakukan berdasarkan skenario yang dijelaskan pada Tabel 4.12.

   Tabel 4.12 Skenario pengujian verifikasi perizinan melalui aplikasi mobile verifikasi perizinan

Skenario	**Jenis Pengujian**	**Keluaran yang diharapkan**
I	Memindai QR code perizinan dengan struktur tidak valid	Aplikasi menampilkan pesan kesalahan “Bukan QR code perizinan”
II	Memindai QR code perizinan dengan *signature* tidak valid	Aplikasi menampilkan pesan kesalahan “Verifikasi tanda tangan digital gagal, QR code perizinan terindikasi palsu”
III	Memindai QR code perizinan dengan data perizinan tidak ditemukan	Aplikasi menampilkan pesan kesalahan “Data perizinan tidak ditemukan, perizinan terindikasi palsu”
IV	Memindai QR code perizinan valid	Aplikasi menampilkan data perizinan

   Berikut penjelasan setiap skenario pengujian:

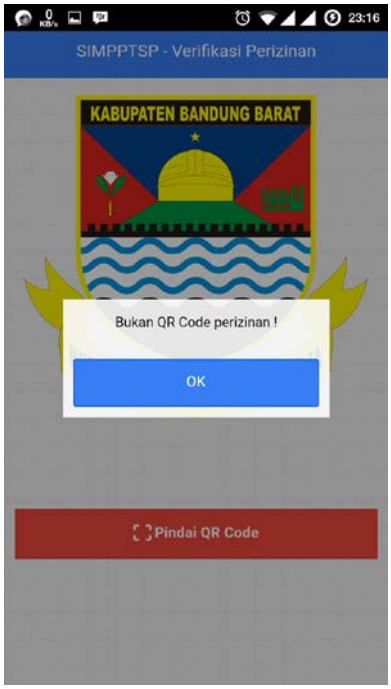
   1. Skenario pengujian I

      Pada skenario pengujian I diasumsikan dilakukan pemalsuan QR code dengan cara menyertakan QR code, namun QR code yang disertakan bukan merupakan QR code perizinan yang valid sehingga tidak memiliki struktur data yang valid, Spesifikasi QR code skenario pengujian I dijelaskan pada Tabel 4.13.

Tabel 4.13 Spesifikasi QR code skenario pengujian I

Data	QR Code
npYscK5KuNom55YE3BGP L8gcTaDuh52jQy5LxSXZlL V2Fknu9Pi6WfxmfMAy5Tao VjpwiuxKKDZKhHO1B2U38 aR3Q3ITr13NlpcqaQbUk7GIZ H1GYyfCA3gLFduQLsAJ	Gambar 4.14 QR code skenario pengujian I

Pengujian dilakukan dengan cara melakukan pemindaian QR code dengan menggunakan aplikasi *mobile* verifikasi perizinan menghasilkan keluaran seperti yang dapat dilihat pada Gambar 4.15.



Gambar 4.15 Keluaran aplikasi mobile verifikasi perizinan saat membaca QR code skenario pengujian I

Keluaran dari aplikasi berupa pesan kesalahan “Bukan QR Code perizinan” , dapat disimpulkan hasil pengujiaan aplikasi *mobile* verifikasi perizinan terhadap skenario pengujian I diterima.

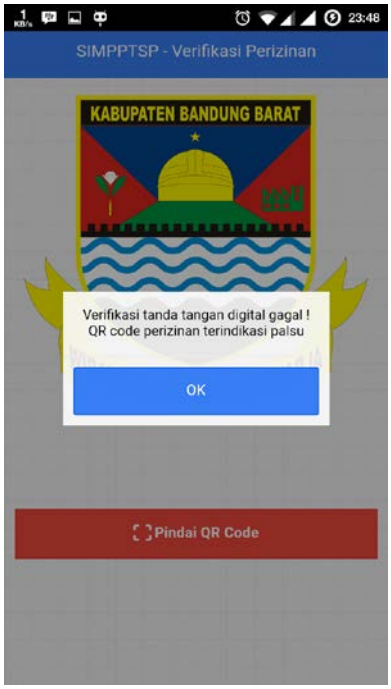
2. Skenario pengujian II

   Pada skenario pengujian II diasumsikan dilakukan pemalsuan QR code dengan cara menyertakan QR code yang memiliki struktur yang valid namun memiliki *signature* yang tidak valid yang artinya data tidak ditandatangani menggunakan *private key* milik BPMPPT KBB atau telah dilakukan perubahan pada data, contoh dari QR code skenario pengujian II dijelaskan pada Tabel 4.14.

Tabel 4.14 Spesifikasi QR code skenario pengujian II

Data	QR Code
MEQCID0WmQrOsNauzyR3l/dTxskUorNnpUKtSz0nwqz4ve9TAiD9Vw1Hl2bZDugEP5uxQC6jHDXfkHDcN0P+Z24Nd3w7Pw==;00408/10-17/PK/X/2016;51029408	 Gambar 4.16 QR code skenario pengujian II

Pengujian dilakukan dengan cara melakukan pemindaian QR code dengan menggunakan aplikasi *mobile* verifikasi perizinan menghasilkan keluaran seperti yang dapat dilihat pada Gambar 4.17.



Gambar 4.17 Keluaran aplikasi mobile verifikasi perizinan saat membaca QR code skenario pengujian II

Keluaran dari aplikasi adalah " Verifikasi tanda tangan digital gagal, QR code perizinan terindikasi palsu ", dapat disimpulkan hasil pengujiaan aplikasi *mobile* verifikasi perizinan terhadap skenario pengujian II diterima.

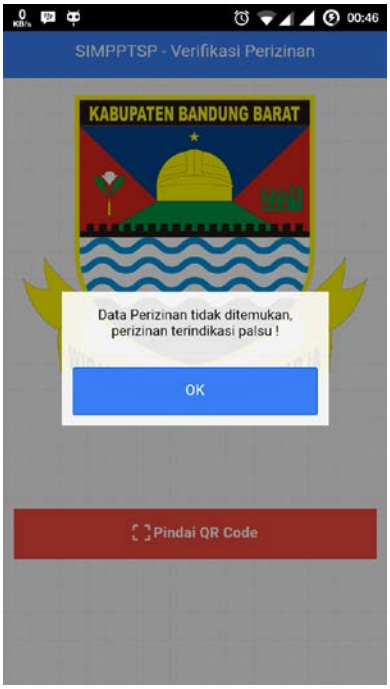
3. Skenario pengujian III

Pada skenario pengujian III diasumsikan dilakukan pemalsuan QR code dengan cara menyertakan QR code yang memiliki struktur yang valid dan *signature* yang valid namun data perizinan tidak ada di basis data SIMPPTSP, contoh dari QR code skenario pengujian III dijelaskan pada Tabel 4.15.

Tabel 4.15 Spesifikasi QR code pengujian skenario III

Data	QR Code
MEQCINxJU5hTCisOQT08QMOdFZlp6zaEjA3Ybf2uTadnh9UFAiDU5OEAJRvHoXd5EcFTzDIxBXWypOoOfQGr/JS4tFyUJQ==;00409/10-17/PK/X/2014;51029409	Gambar 4.18 QR code skenario pengujian III

Pengujian dilakukan dengan cara melakukan pemindaian QR code dengan menggunakan aplikasi *mobile* verifikasi perizinan menghasilkan keluaran seperti yang dapat dilihat pada Gambar 4.19



Gambar 4.19 Keluaran aplikasi mobile verifikasi perizinan saat membaca QR code skenario pengujian III

Keluaran dari aplikasi adalah “Data perizinan tidak dapat ditemukan , perizinan terindikasi palsu“, dapat disimpulkan hasil pengujiaan aplikasi *mobile* verifikasi perizinan terhadap skenario pengujian III diterima.

4. Skenario pengujian IV

Pada skenario pengujian IV diasumsikan dilakukan pemindaian pada QR code perizinan yang memiliki struktur valid, *signature* valid dan data datanya ada di basis data SIMPPTSP yang artinya perizinan valid. Contoh dari QR code skenario pengujian IV dapat dijelaskan pada Tabel 4.16.

Tabel 4.16 Spesifikasi QR code skenario pengujian IV

Data	QR Code
MEQCIG75ST9+ZgkHpM6lWk7UA3VPaUpEedbzxFhIXZnhhAOXAiAKRt3NF7dXPPvYz7E6pk819e5cDmLLPrBmkHawNsWS+A==;00408/10-17/PK/X/2015;51029408	Gambar 4.20 QR code skenario pengujian IV

Pengujian dilakukan dengan cara melakukan pemindaian QR code dengan menggunakan aplikasi *mobile* verifikasi perizinan menghasilkan keluaran seperti yang dapat dilihat pada Gambar 4.21.



Gambar 4.21 Keluaran aplikasi mobile verifikasi perizinan saat membaca QR code skenario pengujian IV

Keluaran dari aplikasi adalah “Data dari perzinan yang diverifikasi “, dapat disimpulkan hasil pengujiaan aplikasi *mobile* verifikasi perizinan terhadap skenario pengujian IV diterima.

Hasil pengujian dengan mempertimbangkan skenario – skenario yang telah dijelaskan pada Tabel 4.17.

Tabel 4.17 Hasil pengujian verifikasi perizinan melalui aplikasi mobile verifikasi perizinan

Skenario	Jenis Pengujian	Pengamatan	Hasil Pengujian
I	Memindai QR code perizinan dengan struktur tidak valid	Aplikasi menampilkan pesan kesalahan “Bukan QR code perizinan”	Diterima
II	Memindai QR code perizinan dengan *signature* tidak valid	Aplikasi menampilkan pesan kesalahan “Verifikasi tanda tangan digital gagal, QR code perizinan terindikasi palsu”	Diterima
III	Memindai QR code perizinan dengan data perizinan tidak ditemukan	Aplikasi menampilkan pesan kesalahan “Data perizinan tidak ditemukan, perizinan terindikasi palsu”	Diterima

IV	Memindai QR code perizinan valid	Aplikasi menampilkan data perizinan	Diterima

5. Pengujian kompatibilitas aplikasi *mobile* verifikasi perizinan

   Pengujiaan dilakukan untuk mengetahui apakah aplikasi *mobile* verifikasi perizinan dapat berjalan sesuai dengan kebutuhan perangkat lunak yang telah ditentukan, pengujian dilakukan dengan menggunakan *emulator* android, sehingga pengujiaan hanya terbatas pada pengujiaan instalasi dan membuka antar muka pemindaian QR code, hasil yang diharapkan dari pengujiaan adalah aplikasi dapat berjalan pada android dari versi 4.1.x dan generasi selanjutnya, hasil dari pengujian yang dilakukan dijelaskan pada Tabel 4.18.

Tabel 4.18 Hasil Pengujian kompatibilitas aplikasi mobile verifikasi perizinan

Versi Android	Codename	Api Level	Hasil Pengujian
4.1.2	Jelly Bean	16	Diterima
4.2.2		17	Diterima
4.3		18	Diterima
4.4	Kitkat	19	Diterima
5.0	Lollipop	21	Diterima
5.1		22	Diterima
6.0	Marshmallow	23	Diterima

4.2.3 Kesimpulan Pengujian Fitur Verifikasi Perizinan Berbasis QR code

Berdasarkan hasil dari pengujian fungsional dengan kasus uji seperti yang telah dijelaskan, dapat ditarik kesimpulan bahwa penambahan fitur verifikasi perizinan berbasis QR code pada SIMPPTSP sudah dapat digunakan dan sesuai dengan tujuan dari penelitian, dengan penjelasan sebagai berikut:

1. Perizinan lebih sulit dipalsukan karena diamankan dengan menggunakan QR code dan *digital signature*.
2. Verifikasi perizinan dapat dilakukan dengan cepat dan tepat baik itu melalui aplikasi *mobile* verifikasi perizinan ataupun halaman *web* verifikasi perizinan.

4.2.4 Rencana Pengujian Pengamanan Dengan TLS

Pada tabel Tabel 4.8 dijelaskan rincian dari rencana pengujiaan fitur verifikasi perizinan berbasis QR code.

Tabel 4.19 Rencana pengujian pengamanan dengan TLS

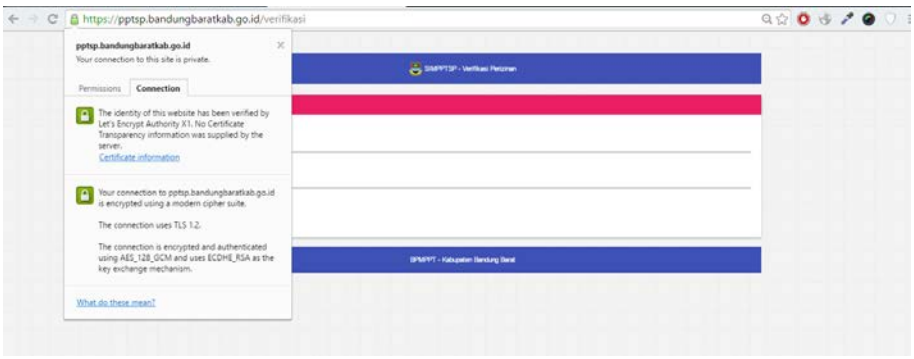
Pengujian	Butir Uji	Jenis Pengujian
Pengamanan dengan menggunakan TLS	Mengakses SIMPPTSP melalui *port* 443 (*https*)	Blackbox
	Mengakses SIMPPTSP melalui *port* 80 (*http*)	Blackbox
	Evaluasi konfigurasi TLS dengan menggunakan *Qualsys* SSL *server test*	Blackbox
	Pengujian kompatibilitas *client* dengan menggunakan *Qualsys SSL server test*	Blackbox
	Perbandingan keamanan sesi komunikasi sebelum dan sesudah diterapkan TLS dengan menggunakan *Wireshark*.	Blackbox

4.2.5 Pengujian Pengamanan Dengan TLS

Pengujian yang dilakukan adalah menguji implementasi dari TLS yang diterapkan pada *web server* SIMPPTSP, hasil dari pengujian tersebut adalah sebagai berikut:

1. Mengakses SIMPPTSP melalui *port* 443

   Pengujian ini dilakukan untuk mengetahui apakah SIMPPTSP dapat diakses melalui *port* 443 (*https)*, gambaran hasil pengujian yang dapat dilihat pada Gambar 4.22 dengan hasil pengujian seperti dijelaskan pada Tabel 4.20.



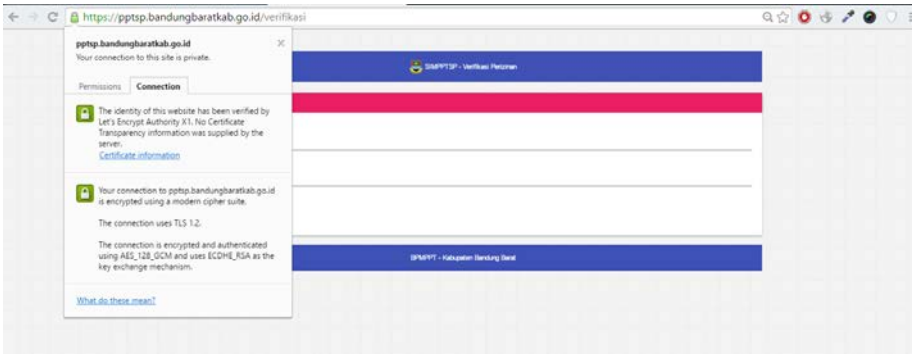
Gambar 4.22 Gambaran hasil pengujian mengakses SIMPPTSP melalui port 443(https)

Tabel 4.20 Hasil pengujian mengakses SIMPPTSP melalui port 443(https)

Pengujian	Keluaran yang Diharapkan	Pengamatan	Hasil
Mengakses SIMPPTSP melalui *port* 443 (*https*)	SIMPPTSP dapat diakses melalui *port* 443 (*https*)	SIMPPTSP dapat diakses melalui *port* 443 (*https*)	Diterima

2. Mengakses SIMPPTSP melalui port 80

   Pengujian ini dilakukan untuk mengetahui apakah SIMPPTSP melakukan pengalihan permintaan dari *port* 80 (*http*) ke *port* 443 (*https*), gambaran hasil pengujian dapat dilihat pada Gambar 4.23, dengan hasil pengujian seperti dijelaskan pada Tabel 4.21.



Gambar 4.23 Gambaran hasil pengujian mengakses SIMPPTSP melalui port 80(http)

Tabel 4.21 Hasil pengujian mengakses SIMPPTSP melalui port 80(http)

Pengujian	Keluaran yang Diharapkan	Pengamatan	Hasil
Mengakses SIMPPTSP melalui *port* 80 (*http*)	SIMPPTSP melakukan pengalihan permintaan dari *port* 80(*http*) ke *port* 443(*https*)	SIMPPTSP melakukan pengalihan permintaan dari *port* 80(*http*) ke *port* 443(*https*)	Diterima

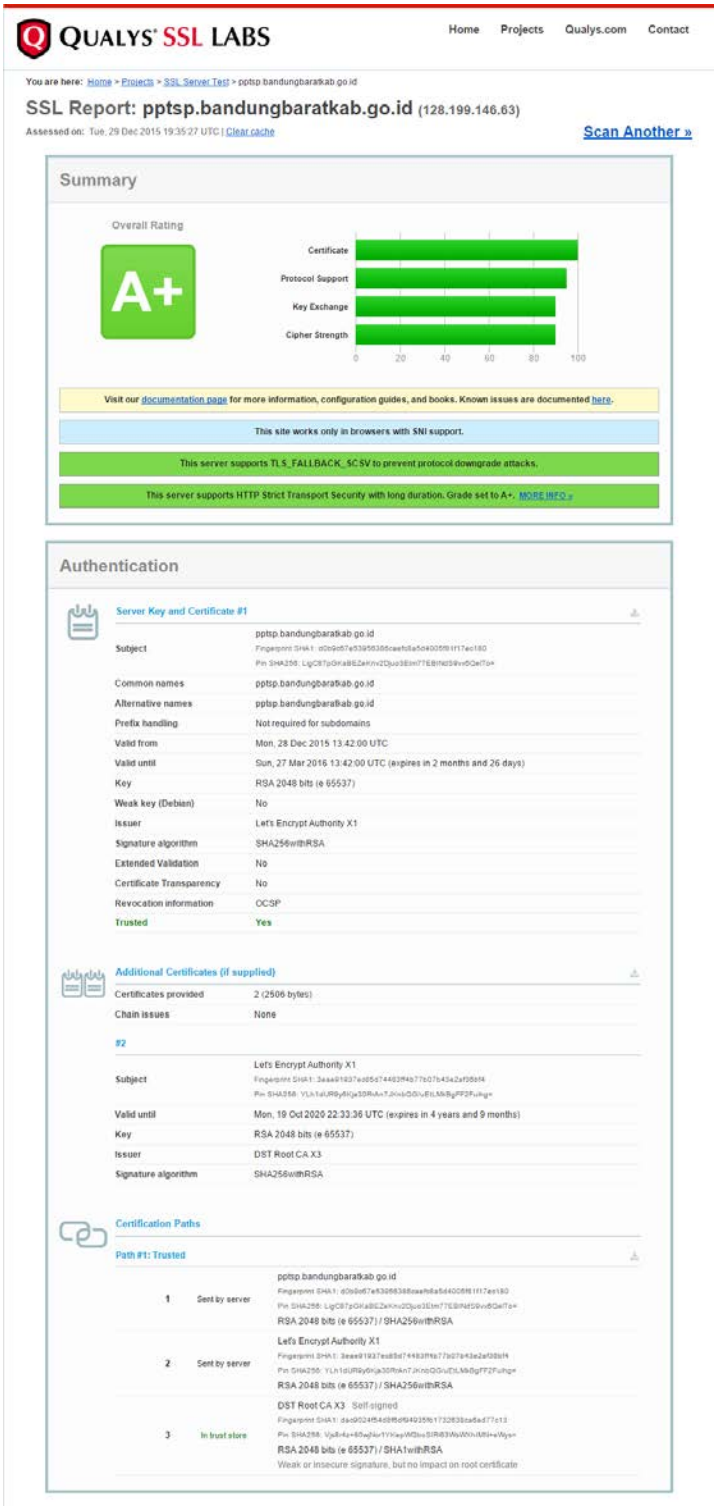
3. Evaluasi konfigurasi TLS dengan menggunakan *Qualsys* SSL *server* test

   Pengujian ini dilakukan untuk evaluasi konfigurasi TLS yang telah di implementasikan, *Qualsys* SSL *server test* adalah layanan yang dapat digunakan untuk evaluasi konfigurasi SSL/TLS dari sebuah *server*, berdasarkan panduan *rating server* yang dikeluarkan oleh *Qualsys* SSL *Labs* [32] ada 8 tingkatan nilai konfigurasi SSL/TLS sebuah *server*, yang dijelaskan pada Tabel 4.22.

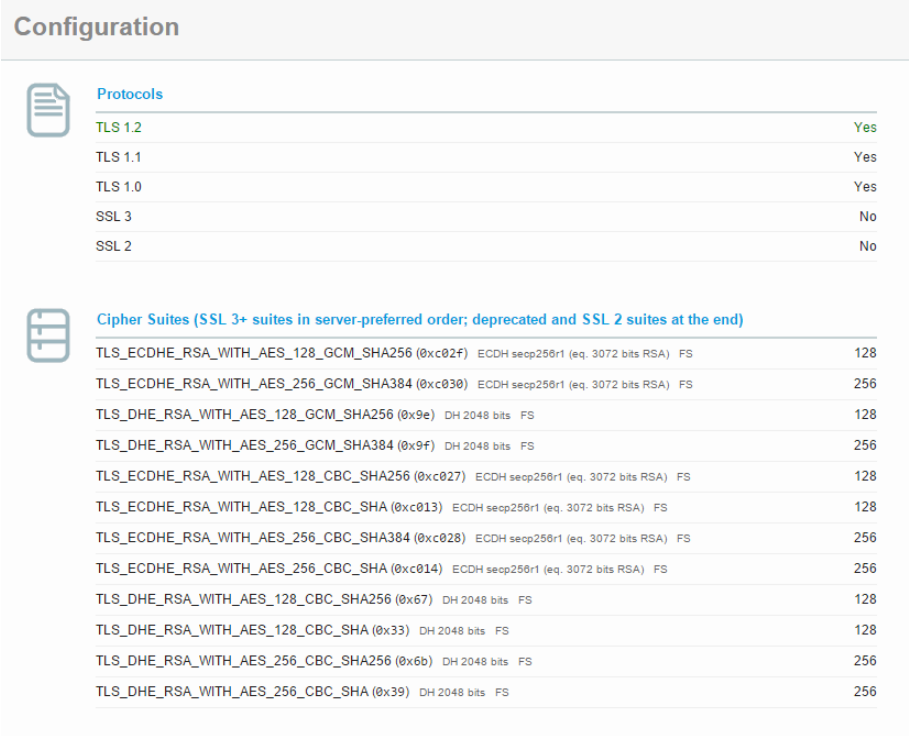
Tabel 4.22 Tingkatan nilai konfigurasi SSL/TLS Qualsys SSL server test

Skor Angka	Tingkatan
Skor >=80	A
Skor>=65	B
Skor>=50	C
Skor>=35	D
Skor>=20	E
Skor<=20	F

Gambaran hasil pengujian dapat dilihat pada Gambar 4.24 dan Gambar 4.25, dengan hasil pSengujian seperti dijelaskan pada Tabel 4.23.



Gambar 4.24 Gambaran hasil pengujian evaluasi konfigurasi TLS dengan menggunakan Qualsys SSL server test I



Gambar 4.25 Gambaran hasil evaluasi konfigurasi TLS dengan menggunakan Qualsys SSL server test II

Tabel 4.23 Hasil pengujian evaluasi konfigurasi TLS dengan menggunakan Qualsys SSL server test

Pengujian	**Keluaran yang Diharapkan**	**Pengamatan**	**Hasil**
Pengujian konfigurasi TLS dengan menggunakan *Qualsys* SSL *server test*	Mendapatkan nilai evaluasi konfigurasi TLS minimal A	Mendapatkan nilai evaluasi konfigurasi TLS A+	Diterima

4. Pengujian kompatibilitas *client* dengan menggunakan *Qualsys SSL server test*

   Pengujian ini dilakukan untuk mengetahui kompatibilitas dari *client* yang akan digunakan untuk mengakses layanan verifikasi perizinan, *Qualsys* SSL *server Test* selain menguji konfigurasi TLS dari sebuah server juga dapat digunakan untuk melakukan simulasi TLS *handshake* sehingga dapat diketahui kompatibilitas dari client yang digunakan, gambaran hasil

pengujiaan dapat dilihat pada Gambar 4.26 , dengan hasil pengujiaan seperti dijelaskan pada Tabel 4.24.



Handshake Simulation

Android 2.3.7 No SNI [2]	Incorrect certificate because this client doesn't support SNI TLS 1.0 TLS_DHE_RSA_WITH_AES_128_CBC_SHA DH 2048
Android 4.0.4	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.1.1	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.2.2	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.3	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.4.2	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 5.0.0	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Baidu Jan 2015	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
BingPreview Jan 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 47 / OS X R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 31.3.0 ESR / Win 7	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 42 / OS X R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Googlebot Feb 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
IE 6 / XP No FS [1] No SNI [2]	Protocol or cipher suite mismatch
IE 7 / Vista	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
IE 8 / XP No FS [1] No SNI [2]	Incorrect certificate because this client doesn't support SNI TLS 1.0 TLS_RSA_WITH_3DES_EDE_CBC_SHA
IE 8-10 / Win 7 R	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
IE 11 / Win 7 R	TLS 1.2 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 11 / Win 8.1 R	TLS 1.2 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 10 / Win Phone 8.0	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
IE 11 / Win Phone 8.1 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
IE 11 / Win Phone 8.1 Update R	TLS 1.2 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 DH 2048 FS
IE 11 / Win 10 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Edge 13 / Win 10 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Edge 13 / Win Phone 10 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Java 6u45 No SNI [2]	Client does not support DH parameters > 1024 bits TLS 1.0 TLS_DHE_RSA_WITH_AES_128_CBC_SHA DH 2048
Java 7u25	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Java 8u31	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 0.9.8y	TLS 1.0 TLS_DHE_RSA_WITH_AES_128_CBC_SHA DH 2048 FS
OpenSSL 1.0.1l R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.0.2 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 5.1.9 / OS X 10.6.8	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Safari 6 / iOS 6.0.1 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 6.0.4 / OS X 10.8.4 R	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Safari 7 / iOS 7.1 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
Safari 9 / iOS 9 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Apple ATS 9 / iOS 9 R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Yahoo Slurp Jan 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
YandexBot Jan 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS

(1) Clients that do not support Forward Secrecy (FS) are excluded when determining support for it.
(2) No support for virtual SSL hosting (SNI). Connects to the default site if the server uses SNI.
(3) Only first connection attempt simulated. Browsers sometimes retry with a lower protocol version.
(R) Denotes a reference browser or client, with which we expect better effective security.
(All) We use defaults, but some platforms do not use their best protocols and features (e.g., Java 6 & 7, older IE).

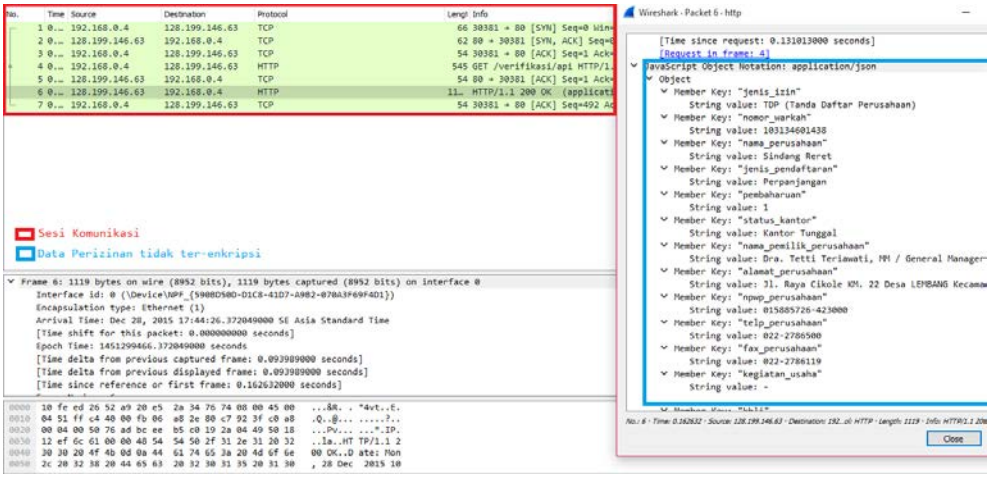
Gambar 4.26 Gambaran hasil pengujian kompatibilitas client dengan menggunakan Qualsys SSL server test

Tabel 4.24 Hasil pengujian kompatibilitas client dengan menggunakan Qualsys SSL server test

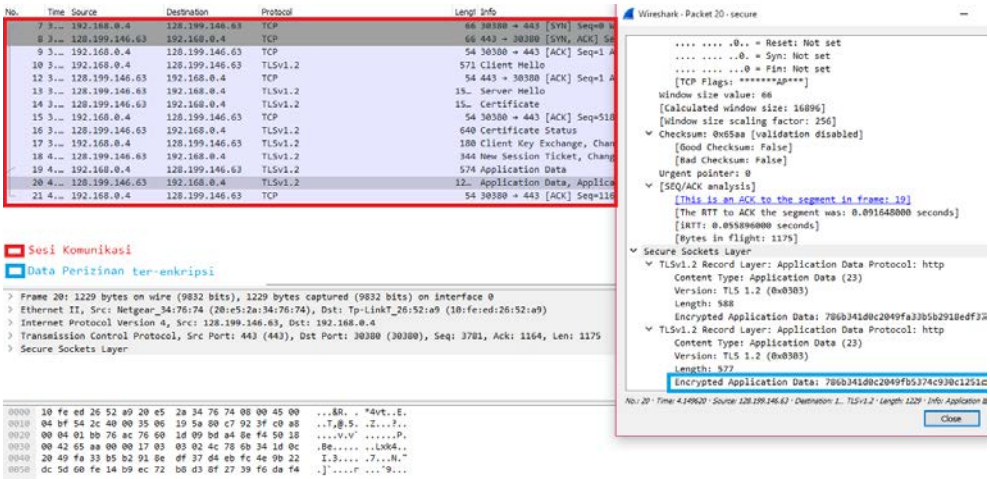
Pengujian	Keluaran yang Diharapkan	Pengamatan	Hasil
Pengujian kompatibilitas *client* dengan menggunakan *Qualsys SSL server test*.	Kompatibel dengan ponsel android dengan sistem operasi android 4.1.x dan generasi penerusnya	Simulasi *handshake* berhasil yang artinya server kompatibel dengan client yang telah ditentukan.	Diterima

5. Perbandingan keamanan sesi komunikasi verifikasi perizinan sebelum dan sesudah diterapkan TLS dengan menggunakan Wireshark.

   Pengujian ini dilakukan untuk mengetahui apakah setelah diterapkan TLS sesi komunikasi dari dan menuju SIMPPTSP lebih aman jika dibandingkan dengan sebelum diterapkan TLS, pengujian dilakukan dengan menggunakan perangkat lunak *wireshark*, *wireshark* dapat digunakan untuk memonitor lalulintas data pada jaringan sehingga dapat terlihat apakah suatu sesi komunikasi ter-enkripsi atau tidak. Dilakukan perbandingan sebelum dan sesudah penerapan TLS dengan gambaran hasil pengujian yang dapat dilihat pada Gambar 4.27 dan Gambar 4.28 , dengan hasil pengujian seperti dijelaskan pada Tabel 4.25.



Gambar 4.27 Gambaran sesi komunikasi sebelum diterapkan TLS



Gambar 4.28 Gambaran sesi komunikasi setelah diterapkan TLS

Tabel 4.25 Hasil pengujian perbandingan keamanan sesi komunikasi sebelum dan sesudah diterapkan TLS dengan menggunakan wireshark

Pengujian	Keluaran yang Diharapkan	Pengamatan	Hasil
Perbandingan keamanan sesi komunikasi sebelum dan sesudah diterapkan TLS dengan menggunakan *Wireshark.*	Sesi komunikasi lebih aman dari sebelum diterapkan TLS	Sesi komunikasi verifikasi perizinan ter-enkripsi	Diterima

4.2.6 Kesimpulan Pengujian Pengamanan Dengan TLS

Berdasarkan hasil dari pengujian dengan kasus uji seperti yang telah dijelaskan, dapat ditarik kesimpulan bahwa pengamanan sesi komunikasi yang diterapkan pada *web server* SIMPPTSP sudah berjalan dengan baik dan sesuai dengan tujuan dari penelitian, hal ini terlihat dari hasil pengujian yang telah dilakukan dimana konfigurasi TLS yang diterapkan mendapatkan evaluasi A+ menurut standar *Qualsys SSL labs*, serta terenkripsinya seluruh sesi komunikasi dari dan menuju SIMPPTSP.