# **BAB IV**

**PENGUJIAN DAN IMPLEMENTASI**

## **Pengujian Sistem**

Teori

### ***Blackbox Testing***

Teori

### ***Whitebox Testing***

Teori umum

Tabel Whitebox Testing Input Data Bus dengan Vinegere Cipher

|  |  |
| --- | --- |
| 1 | public function add\_bus(){  $nomor\_plat\_kendaraan = $this->input->post('nomor\_plat\_kendaraan', true);  $jenis\_angkutan = $this->input->post('jenis\_angkutan', true);  $nama\_po = $this->input->post('nama\_po', true);  $trayek = $this->input->post('trayek', true); |
| 2 | if (isPlatExist($nomor\_plat\_kendaraan)) {  return responseBAD("Nomor Plat Kendaraan ini '" . $nomor\_plat\_kendaraan . "' Sudah terpakai !");  } |
| 3 | $id\_bus = vigenereCipher($nomor\_plat\_kendaraan, 'encrypt'); |
| 4 | function vigenereCipher($inputText, $mode = 'encrypt'){  $key = VIGENERE\_CIPHER\_KEY;  $result = '';  $inputText = strtoupper($inputText);  $key = strtoupper($key);  $keyLength = strlen($key);  $inputLength = strlen($inputText); |
| 5 | for ($i = 0; $i < $inputLength; $i++) {  $inputChar = ord($inputText[$i]);  $keyChar = ord($key[$i % $keyLength]); |
| 6 | if ($inputChar >= 65 && $inputChar <= 90) {  if ($mode === 'encrypt') {  $encryptedChar = ($inputChar + $keyChar - 130) % 26 + 65;  } else {  $encryptedChar = ($inputChar - $keyChar + 26) % 26 + 65;  }  $result .= chr($encryptedChar); }  else {  $result .= $inputText[$i];  }  } |
| 7 | return removeNonAlphanumeric($result);  } |
| 9 | $data = array(  'id\_bus' => $id\_bus,  'nomor\_plat\_kendaraan' => strtoupper($nomor\_plat\_kendaraan),  'jenis\_angkutan' => $jenis\_angkutan,  'nama\_po' => $nama\_po,  'trayek' => $trayek  );  $this->Global\_model->insert('master\_bus', $data); |
| 10 | $this->load->library('ciqrcode');  $fill = $id\_bus;  $config['cacheable'] = true;  $config['cachedir'] = './assets/';  $config['errorlog'] = './assets/';  $config['imagedir'] = '.' . QR\_PATH;  $config['quality'] = true;  $config['size'] = '1024';  $config['black'] = array(224, 255, 255);  $config['white'] = array(70, 130, 180);  $this->ciqrcode->initialize($config);  $image\_name = $id\_bus . '.png';  $params['data'] = $fill;  $params['level'] = 'H';  $params['size'] = 10;  $params['savename'] = FCPATH . $config['imagedir'] . $image\_name;  $this->ciqrcode->generate($params); |
| 11 | responseOK("Berhasil"); |
| 12 | } |

Keterangan:

|  |  |  |
| --- | --- | --- |
| Node 1 | : | Pemanggilan publik fungsi tambah bus, inisialisasi variabel menerima data dari variabel view. |
| Node 2 | : | Kondisi respon erorr jika plat nomor sudah ada ada di database. |
| Node 3 | : | Memanggil publik fungsi vigenereCipher pada helper |
| Node 4 | : | Pada public fungsi vigenereCipher dilakukan inisiali variabel untuk algoritma vigenereCipher |
| Node 5 | : | Kemudian, fungsi melakukan loop melalui setiap karakter dalam teks input dengan menggunakan loop. |
| Node 6 | : | Fungsi kemudian memeriksa apakah karakter dalam teks input berada dalam rentang huruf kapital (ASCII 65-90), yang sesuai dengan huruf A hingga Z dalam alfabet. Jika ya, itu adalah karakter yang akan dienkripsi atau didekripsi. Jika tidak, karakter tersebut akan disalin langsung ke hasil tanpa perubahan. Selanjutnya, tergantung pada mode ('encrypt' atau 'decrypt'), karakter dalam teks input dienkripsi atau didekripsi menggunakan rumus Vigenere Cipher. Proses ini melibatkan perhitungan yang memanfaatkan pergeseran karakter dan aritmatika modular. Hasil karakter terenkripsi atau didekripsi kemudian diubah kembali menjadi karakter dengan menggunakan chr() dan ditambahkan ke hasil akhir. |
| Node 7 | : | Akhirnya, hasil enkripsi atau dekripsi akan melewati fungsi removeNonAlphanumeric() sebelum dikembalikan oleh fungsi vigenereCipher($nomor\_plat\_kendaraan, 'encrypt'); |
| Node 8 | : | Penyimpanan data bus kedalam kedatabse |
| Node 9 | : | Dari hasil kode bus yang telah di acak oleh Vigenere Cipher, dilakukan generate ke QR Code dan disimpan ke dalam asset berupa gambar. |
| Node 10 | : | Kondisi respon sukses |
| Node 11 | : | Selesai |



**Gambar** *Flowgraph Nation*

Perhitungan Flowgraph Nation

V(G) = E – N + 2

Dimana :

E (jumlah garis) = 11

N (jumlah node) = 11

Jadi, V(G) = 11 – 11 + 2

V(G) = 2

Cyclomatyc Complexity

R1, R2 = 2

Jadi, Cyclomatic Complexity untuk Flowgraph ini = 2

Langkah alur Cyclomatic Complexity adalah sebagai berikut :

Path 1 : 1-2-11

Path 2 : 1-2-3-4-5-6-7-8-9-10-11

Tabel 4. 5 Alur Penguian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Langkah Alur** | **Cara Menguji** | **Hasil Yang Diharapkan** | **Hasil Yang Dikeluarkan** | **Ket** |
| 1-2-3-11 | Admin melakukan input data bus yang sebelumnya plat nomer telah terdaftar di sistem. | Data bus yang dimasukan terutama data plat nomor akan dilakukan pengecekan data di dalam database, jika terdapat data yang sama didalam database. Maka sistem akan memberikan respon eror, dan menampilkan pop up erorr ”Plat kendaraan sudah terpakai, dan tidak menyimpan data. | Sesuai harapan | Valid |
| 1-2-3-4-5-6-11 | Admin melakukan input data bus yang sebelumnya plat nomer tidak terdaftar di sitem. | Data bus yang dimasukan terutama data plat nomor akan dilakukan pengecekan data di dalam database, jika tidak terdapat data yang sama. Maka akan dilakukan pengacakan ID bus dengan algoritma Vigenere Cipher, selanjutnya hasil pangacakan ID Bus tersebut akan diganrate berupa QR Code, data tersimpan di database dan sistem memberikan respone sukses yang menampilkan pop up sukses ”Data Berhasil Disimpan” | Sesuai harapan | Valid |

1. **Implementasi Sistem**

Teori

1. **Implementasi Antar Muka Android**

Teori

1. **Implementasi Antar Muka WEB**

Teori