

**CAK3BAB3 IMPLEMENTASI DAN PENGUJIAN PERANGKAT
LUNAK**

WHITE BOX TESTING



ANGGOTA KELOMPOK :

Ario Mukti elsandy	(2211102184)
Muhamad Nu'man alkafi	(2211102223)
Rafli Kurniawan	(2211102201)

DOSEN :

Muhamad Lulu Latif Usman, S.Pd., M.Han

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
PURWOKERTO**

2024

WHITE BOX TESTING

1. Tabel Identitas Penguji

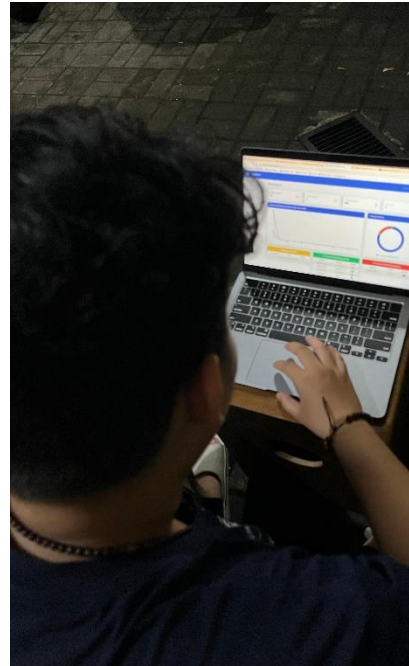
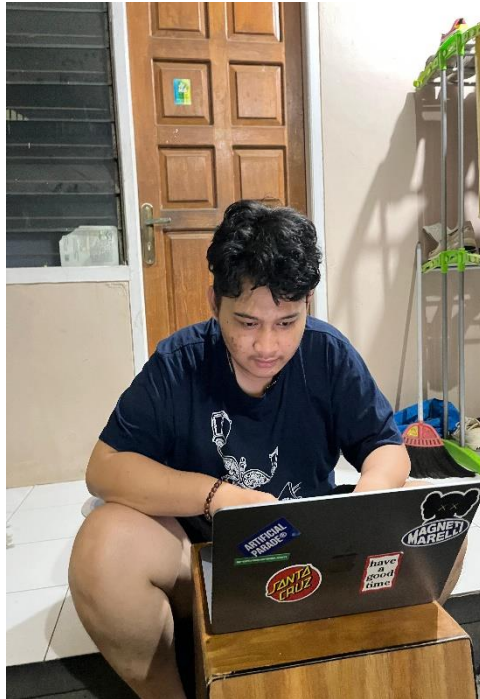
No	Nama Penguji	Umur	Jenis Kelamin	Pekerjaan
1	Muhamad adi pratama	21	L	Mahasiswa teknik telekomunikasi

2. Tabel Rincian Fitur yang Diuji

No	Fitur yang Diuji	Deskripsi Fitur	Tujuan Pengujian	Harapan Hasil
1	Pendaftaran akun	Fungsi Pendaftaran akun	Memastikan fungsi pendaftaran berjalan dengan benar	Berhasil mendaftarkan akun
2	Login Admin	Fungsi Login Admin	Memastikan fungsi login berjalan dengan benar	Berhasil login
3	Pengaktifan akun oleh admin			
4	Tambah Distributor	Fungsi Tambah Distributor	Memastikan Distributor baru dapat ditambahkan dengan benar	Distributor berhasil ditambah
5	Edit Distributor	Fungsi Edit Distributor	Memastikan Distributor baru dapat diedit dengan benar	Distributor berhasil diedit
6	Hapus Distributor	Fungsi Hapus Distributor	Memastikan Distributor dapat dihapus dengan benar	Pesanan berhasil dihapus
7	Tambah Satuan barang	Fungsi Tambah Satuan barang	Memastikan Satuan barang dapat ditambahkan dengan benar	Satuan barang berhasil dibuat
8	Edit Satuan barang	Fungsi Edit Satuan barang	Memastikan Satuan barang dapat diedit dengan benar	Satuan barang berhasil diedit
9	Hapus Satuan barang	Fungsi hapus Satuan barang	Memastikan Satuan barang dapat dihapus dengan benar	Satuan barang berhasil dihapus
10	Tambah Jenis barang	Fungsi Tambah Jenis barang	Memastikan Jenis barang dapat ditambahkan dengan benar	Jenis barang berhasil ditambah
11	Edit Jenis barang	Fungsi Edit Jenis barang	Memastikan Jenis barang dapat diedit dengan benar	Jenis barang berhasil diedit
12	Hapus Jenis barang	Fungsi hapus Satuan barang	Memastikan Jenis barang dapat dihapus dengan benar	Jenis barang berhasil dihapus
13	Tambah data barang	Fungsi Tambah data barang	Memastikan data barang dapat ditambahkan dengan benar	Data barang berhasil ditambah

14	Edit data barang	Fungsi Edit data barang	Memastikan data barang dapat diedit dengan benar	data barang berhasil diedit
15	Hapus data barang	Fungsi hapus Satuan barang	Memastikan data barang dapat dihapus dengan benar	Data barang berhasil dihapus
16	Tambah Barang masuk	Fungsi Tambah Barang masuk	Memastikan Barang masuk dapat ditambah dengan benar	Barang masuk berhasil ditambah
17	Hapus Barang masuk	Fungsi hapus Barang masuk	Memastikan Barang masuk dapat dihapus dengan benar	Barang masuk berhasil dihapus
18	Tambah Barang keluar	Fungsi Tambah Barang keluar	Memastikan Barang keluar dapat ditambah dengan benar	Barang keluar berhasil ditambah
19	Hapus Barang keluar	Fungsi hapus Barang keluar	Memastikan Barang keluar dapat dihapus dengan benar	Barang keluar berhasil dihapus
20	Cetak laporan masuk	Fungsi Cetak laporan masuk	Memastikan Cetak laporan dengan benar	Cetak laporan masuk berhasil
21	Cetak laporan keluar	Fungsi Cetak laporan keluar	Memastikan Cetak laporan dengan benar	Cetak laporan masuk berhasil

3. Tabel Pengujian Unit Test dan Tabel Validasi Input



No	Deskripsi Singkat	Jumlah Test Case	Status	
			Passed	Failed
1	Memastikan Pendaftaran akun berhasil			
2	Login admin	3	2	1
3	Pengaktifan akun oleh admin			
4	Memastikan Distributor baru dapat ditambahkan dengan benar			
5	Memastikan Distributor baru dapat diedit dengan benar	3	3	0
6	Memastikan Distributor dapat dihapus dengan benar			
7	Memastikan Satuan barang dapat ditambahkan dengan benar			
8	Memastikan Satuan barang dapat diedit dengan benar	3	3	0
9	Memastikan Satuan barang dapat dihapus dengan benar			
10	Memastikan Jenis barang dapat ditambah dengan benar			
11	Memastikan Jenis barang dapat diedit dengan benar	3	3	0
12	Memastikan Jenis barang			

	dapat dihapus dengan benar			
13	Memastikan data barang dapat ditambah dengan benar			
14	Memastikan data barang dapat diedit dengan benar	3	3	0
15	Memastikan data barang dapat dihapus dengan benar			
16	Memastikan Barang masuk dapat ditambah dengan benar	2	2	0
17	Memastikan Barang masuk dapat dihapus dengan benar			
18	Memastikan Barang keluar dapat ditambah dengan benar	2	2	0
19	Memastikan Barang keluar dapat dihapus dengan benar			
20	Memastikan Cetak laporan masuk dengan benar	2	2	0
21	Memastikan Cetak laporan keluar dengan benar			

4. Kode Testing

LoginTest.php

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Auth extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->library('form_validation');
        $this->load->model('Auth_model', 'auth');
        $this->load->model('Admin_model', 'admin');
    }

    private function _has_login()
    {
        if ($this->session->has_userdata('login_session')) {
            redirect('dashboard');
        }
    }

    public function index()
    {
        $this->_has_login();

        $this->form_validation->set_rules('username', 'Username', 'required/trim');
        $this->form_validation->set_rules('password', 'Password', 'required/trim');

        if ($this->form_validation->run() == false) {
            $data['title'] = 'Login Aplikasi';
            $this->template->load('templates/auth', 'auth/login', $data);
        } else {
            $input = $this->input->post(null, true);

            $cek_username = $this->auth->cek_username($input['username']);
            if ($cek_username > 0) {
                $password = $this->auth->get_password($input['username']);
                if (password_verify($input['password'], $password)) {
                    $user_db = $this->auth->userdata($input['username']);
                    if ($user_db['is_active'] != 1) {
                        set_pesan('akun anda belum aktif/dinonaktifkan. Silahkan hubungi admin.', false);
                        redirect('login');
                    } else {
                        $userdata = [
                            'user' => $user_db['id_user'],
                            'role' => $user_db['role'],
                            'timestamp' => time()
                        ];
                        $this->session->set_userdata('login_session', $userdata);
                        redirect('dashboard');
                    }
                } else {
                    set_pesan('password salah', false);
                    redirect('auth');
                }
            }
        }
    }
}
```

```

        } else {
            set_pesan('username belum terdaftar', false);
            redirect('auth');
        }
    }
}

public function logout()
{
    $this->session->unset_userdata('login_session');

    set_pesan('anda telah berhasil logout');
    redirect('auth');
}

public function register()
{
    $this->form_validation->set_rules('username', 'Username',
    'required/trim/is_unique[user.username]/alpha_numeric');
    $this->form_validation->set_rules('password', 'Password', 'required/min_length[3]/trim');
    $this->form_validation->set_rules('password2', 'Konfirmasi Password',
    'matches[password]/trim');
    $this->form_validation->set_rules('nama', 'Nama', 'required/trim');
    $this->form_validation->set_rules('email', 'Email',
    'required/trim/valid_email/is_unique[user.email]');
    $this->form_validation->set_rules('no_telp', 'Nomor Telepon', 'required/trim');

    if ($this->form_validation->run() == false) {
        $data['title'] = 'Buat Akun';
        $this->template->load('templates/auth', 'auth/register', $data);
    } else {
        $input = $this->input->post(null, true);
        unset($input['password2']);
        $input['password'] = password_hash($input['password'], PASSWORD_DEFAULT);
        $input['role'] = 'gudang';
        $input['foto'] = 'user.png';
        $input['is_active'] = 0;
        $input['created_at'] = time();

        $query = $this->admin->insert('user', $input);
        if ($query) {
            set_pesan('daftar berhasil. Selanjutnya silahkan hubungi admin untuk mengaktifkan akun
anda.');
```

Dashboardt.php

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Dashboard extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();
        $this->load->model('Admin_model', 'admin');
    }

    public function index()
    {
        $data['title'] = "Dashboard";
        $data['barang'] = $this->admin->count('barang');
        $data['barang_masuk'] = $this->admin->count('barang_masuk');
        $data['barang_keluar'] = $this->admin->count('barang_keluar');
        $data['distributor'] = $this->admin->count('distributor');
        $data['user'] = $this->admin->count('user');
        $data['stok'] = $this->admin->sum('barang', 'stok');
        $data['barang_min'] = $this->admin->getBarangMin();
        $data['transaksi'] = [
            'barang_masuk' => $this->admin->getBarangMasukWithDetails(),
            'barang_keluar' => $this->admin->getBarangKeluar(5)
        ];

        // Line Chart
        $b1n = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12'];
        $data['cbm'] = [];
        $data['cbk'] = [];

        foreach ($b1n as $b) {
            $data['cbm'][$b] = $this->admin->chartBarangMasuk($b);
            $data['cbk'][$b] = $this->admin->chartBarangKeluar($b);
        }

        $this->template->load('templates/dashboard', 'dashboard', $data);
    }
}
```


TiketTest.php

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Jenis extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();

        $this->load->model('Admin_model', 'admin');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['title'] = "Jenis";
        $data['jenis'] = $this->admin->get('jenis');
        $this->template->load('templates/dashboard', 'jenis/data', $data);
    }

    private function _validasi()
    {
        $this->form_validation->set_rules('nama_jenis', 'Nama Jenis', 'required|trim');
    }

    public function add()
    {
        $this->_validasi();

        if ($this->form_validation->run() == false) {
            $data['title'] = "Jenis";
            $this->template->load('templates/dashboard', 'jenis/add', $data);
        } else {
            $input = $this->input->post(null, true);
            $insert = $this->admin->insert('jenis', $input);
            if ($insert) {
                set_pesan('data berhasil disimpan');
```

```

        redirect('jenis');
    } else {
        set_pesan('data gagal disimpan', false);
        redirect('jenis/add');
    }
}

public function edit($getId)
{
    $id = encode_php_tags($getId);
    $this->_validasi();

    if ($this->form_validation->run() == false) {
        $data['title'] = "Jenis";
        $data['jenis'] = $this->admin->get('jenis', ['id_jenis' => $id]);
        $this->template->load('templates/dashboard', 'jenis/edit', $data);
    } else {
        $input = $this->input->post(null, true);
        $update = $this->admin->update('jenis', 'id_jenis', $id, $input);
        if ($update) {
            set_pesan('data berhasil disimpan');
            redirect('jenis');
        } else {
            set_pesan('data gagal disimpan', false);
            redirect('jenis/add');
        }
    }
}

public function delete($getId)
{
    $id = encode_php_tags($getId);
    if ($this->admin->delete('jenis', 'id_jenis', $id)) {
        set_pesan('data berhasil dihapus. ');
    } else {
        set_pesan('data gagal dihapus.', false);
    }
    redirect('jenis');
}

```

j
