

HASIL KERJA FR.IA.02. TUGAS PRAKTIK DEMONSTRASI

Skema Sertifikasi (KKNI/Okupasi/Klaster)	Judul	:	Analisis Program
	Nomor	:	SKM-2018-62010-001
TUK		:	Sewaktu/Tempat Kerja/Mandiri*
Nama Asesor		:	Mari Rahmawati, M.Kom
Nama Asesi		:	Muhammad Khaeril Anwar
Tanggal		:	14 Januari 2025

Unit Kompetensi	Kode Unit	:	J. 620100.002.01
	Judul Unit	:	Menganalisis Skalabilitas Perangkat Lunak

Langkah Kerja

- **Mengumpulkan Kebutuhan Skalabilitas**


- **1.1 Identifikasi lingkup (scope) sistem.**
 - Sistem meliputi aplikasi web untuk agen dan petugas PLN, yang memungkinkan mereka mengelola data pelanggan, penggunaan listrik, pembayaran, dan laporan.
- **1.2 Identifikasi lingkungan operasi aplikasi.**
 - **Hardware** : RAM 8 GB, Processor Intel Core I3
 - **Software**: Aplikasi web Browser, Database, Bahasa Pemrograman PHP dengan framework Laravel, Laragon, Windows 11

- **Menganalisis Kebutuhan Skalabilitas**

- **2.1 Analisis masalah skalabilitas berdasar lingkup dan lingkungan operasi sistem.**
 - **Data Volume**: Meningkatnya jumlah pelanggan dan transaksi memerlukan database yang scalable.
 - **Performance**: Server harus mampu menangani banyak request simultan dari agen dan petugas tanpa penurunan kinerja.
- **2.2 Analisis kompleksitas aplikasi sesuai dengan kebutuhan pemrosesan dan jumlah data/pengguna yang akan terlibat.**
 - **Kompleksitas Pemrosesan**: Setiap transaksi harus menghitung biaya berdasarkan tarif, yang memerlukan pemrosesan yang efisien.
 - **Jumlah Pengguna**: Aplikasi harus mampu menangani banyak pengguna aktif secara bersamaan.
- **2.3 Analisis kebutuhan perangkat keras.**
 - **Laptop/PC**
 - **Procesor Intel core i3 gen 10**
 - **Ram 8 Gb**
 - **Storage 1 TB**

- Laragon
- 2.4 Dokumentasikan hasil analisis.

Laragon Full 6.0 220916 php-8.3.13-Win32-vs16-x64 [TS] 192.168.240.190



Menu

Apache Apache24 started80Reload

MySQL mysql-8.0.30-winx64 started3306

StopWebDatabaseTerminalRoot

Device specificationsCopy

Device nameDESKTOP-VRIADEE

ProcessorIntel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz

Installed RAM8.00 GB (7.87 GB usable)

Device IDD6B6C588-C73D-4115-95DB-4DFA89BD65D8

Product ID00330-80000-00000-AA273

System type64-bit operating system, x64-based processor

Pen and touchNo pen or touch input is available for this display

Related linksDomain or workgroupSystem protectionAdvanced system settings

Windows specificationsCopy

EditionWindows 11 Pro

Version23H2

Installed on03/11/2024

OS build22631.4602

ExperienceWindows Feature Experience Pack 1000.22700.1055.0

Microsoft Services Agreement

Microsoft Software Licence Terms

Unit Kompetensi	Kode Unit	:	J.620100.022.02
	Judul Unit	:	Mengimplementasikan Algoritma Pemrograman

Langkah Kerja

1. Menjelaskan varian dan invarian

1.1. Jelaskan tipe data yang sesuai kaidah pemrograman.

- Integer : tipe data berbentuk bilangan bulat atau tipe data numerik yang umum digunakan untuk menyimpan angka tanpa komponen pecahan dengan rentang angka -707, 0, hingga 707.
- Float : tipe data yang digunakan untuk angka dengan desimal. Cocok untuk perhitungan dengan nilai presisi tinggi.
- Complex : tipe data yang digunakan untuk menyimpan bilangan kompleks. Berguna dalam konteks matematika atau fisika.
- Boolean : tipe data yang digunakan untuk menyimpan nilai logika benar atau salah. Cocok untuk kondisi atau status.
- Array (list) : tipe data berbentuk daftar yang menyimpan sejumlah elemen dalam urutan tertentu dari semua tipe data yang sama.
- String : tipe data ini terdiri dari satu atau lebih karakter yang dapat mencakup huruf, angka, dan jenis karakter lainnya seperti halnya teks biasa

1.2. Jelaskan variabel data yang sesuai kaidah pemrograman.

- Nama variabel tidak boleh didahului dengan symbol dan angka
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada pada Bahasa Python
- Nama variabel bersifat case sensitive
- Disarankan menggunakan underscore untuk nama variabel yang lebih dari 2 suku kata

1.1. Jelaskan konstanta yang sesuai kaidah pemrograman.

Konstanta adalah identifier yang terkait nilai tidak bisa biasanya diubah oleh program selama pelaksanaannya.

2. Membuat alur logika (flowchart)

2.1. Metode yang sesuai ditentukan.

Flowchart Sistem dan Program

2.2. Komponen yang dibutuhkan ditentukan.



= digunakan untuk mulai atau start



= digunakan untuk proses



= digunakan untuk kotak keputusan/percabangan

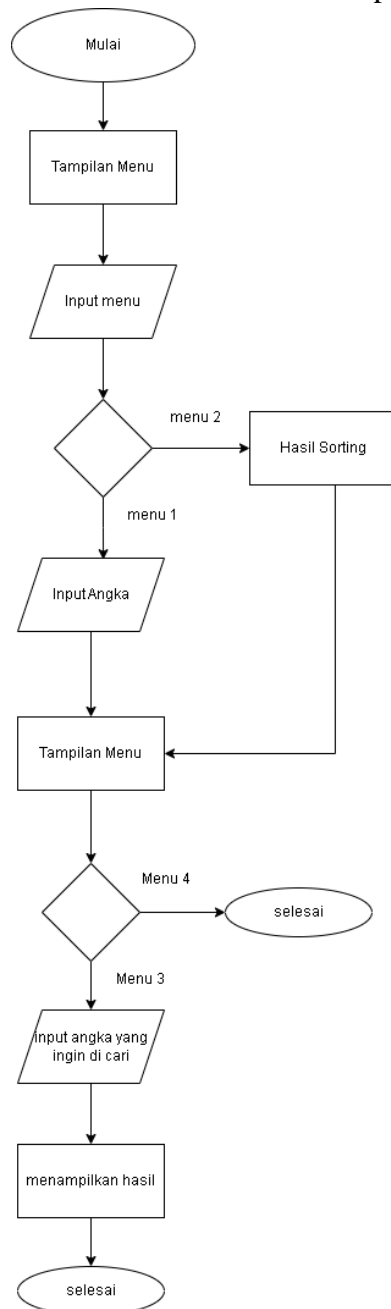


= digunakan untuk input secara manual

2.3. Relasi antar komponen ditetapkan.

Relasi Database adalah model yang paling sederhana dari suatu komponen basisdata yang saling berhubungan antara suatu tabel basis data yang satu dengan tabelbasis data yang lain sehingga mudah untuk dipahami. Model ini menggunakansekumpulan tabel berdimensi dua yaitu relasi dan tabel. Masing – masing relasitersusun atas tupel atau barisan dan atribut. Relasi dirancang sedemikian rupasehingga dapat menghilangkan kemubaziran data redudansi dan menggunakankunci utama atau biasa disebut dengan primary key sebagai penghubung relasi.

2.4. Alur mulai dan selesai ditetapkan.



3. Menerapkan teknik dasar algoritma umum

3.1. Membuat algoritma untuk sorting.

```
def input_numbers():
    numbers = []
    count = int(input("Masukkan jumlah nilai tugas: "))
    print("Input Angka Secara Acak")
```

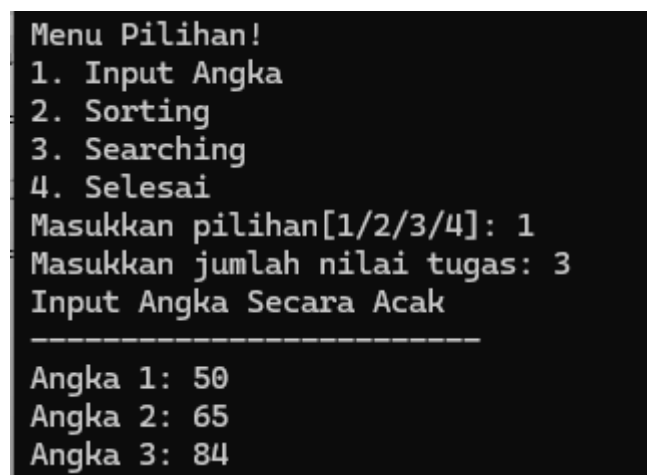
```

print(25*" -")
for i in range(count):
    input_number = int(input(f"Angka {i+1}: "))
    numbers.append(input_number)
return numbers

def sort_numbers(numbers):
    sorted_numbers = sorted(numbers)
    return ", ".join(map(str, sorted_numbers))

while True:
    print("Menu Pilihan!")
    print("1. Input Angka")
    print("2. Sorting")
    print("3. Searching")
    print("4. Selesai")
    choice = int(input("Masukkan pilihan[1/2/3/4]: "))
    if choice == 1:
        numbers = input_numbers()
    elif choice == 2:
        print(f"Hasil sorting : {sort_numbers(numbers)}")
    elif choice == 3:
        search_number()
    elif choice == 4:
        break

```



```

Menu Pilihan!
1. Input Angka
2. Sorting
3. Searching
4. Selesai
Masukkan pilihan[1/2/3/4]: 1
Masukkan jumlah nilai tugas: 3
Input Angka Secara Acak
-----
Angka 1: 50
Angka 2: 65
Angka 3: 84

```

3.2. Membuat algoritma untuk searching.

```

def search_number():
    search_number = int(input("Masukkan angka yang dicari: "))
    if search_number in numbers:
        print(f"Angka ditemukan!")
    else:
        print(f"Angka tidak ditemukan!")

```

```

Menu Pilihan!
1. Input Angka
2. Sorting
3. Searching
4. Selesai
Masukkan pilihan[1/2/3/4]: 3
Masukkan angka yang dicari: 65
Angka ditemukan!

```

4. Menggunakan prosedur dan fungsi

4.1. Identifikasi konsep penggunaan kembali prosedur dan fungsi.

- Fungsi: Fungsi merupakan suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian).
- Prosedur: merupakan suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (program bagian). Prosedur biasanya bersifat suatu aktivitas seperti mencari bilangan prima dari sekumpulan bilangan atau mencari bilangan genap dari sekumpulan bilangan, dsb.

4.2. Gunakan prosedur.

```

def input_numbers():
    numbers = []
    count = int(input("Masukkan jumlah nilai tugas: "))
    print("Input Angka Secara Acak")
    print(25*" -")
    for i in range(count):
        input_number = int(input(f"Angka {i+1}: "))
        numbers.append(input_number)
    return numbers

```

4.3. Gunakan fungsi.

```

def sort_numbers(numbers):
    sorted_numbers = sorted(numbers)
    return ", ".join(map(str, sorted_numbers))

```

5. Mengidentifikasi kompleksitas algoritma

5.1. Identifikasi kompleksitas waktu algoritma.

Kompleksitas waktu algoritma dihitung berdasarkan jumlah operasi perbandingan elemen larik ($A[i] > \text{maks}$). Kompleksitas waktu

Cari Elemen Terbesar : $T(n) = n - 1$

5.2. Identifikasi kompleksitas penggunaan memory algoritma.

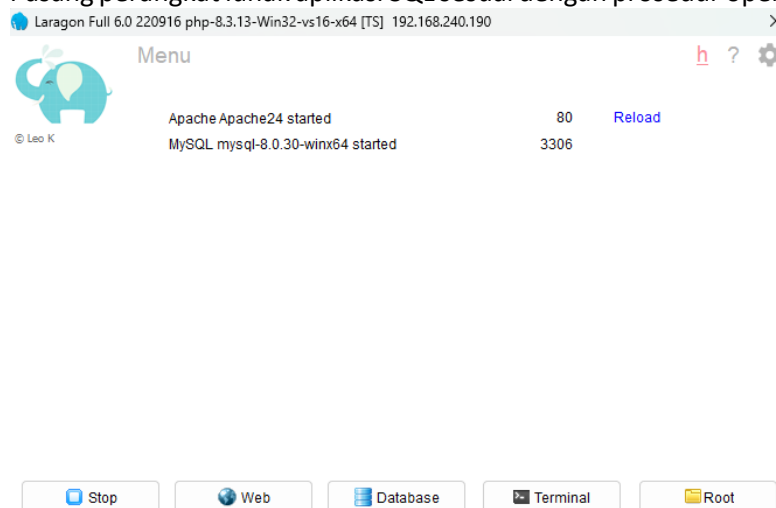
Algoritma yang bagus adalah algoritma yang efisien. Efisiensi suatu algoritma

diukur dari berapa jumlah waktu dan ruang (space) memori yang dibutuhkan untuk menjalankannya. Algoritma yang efisien adalah algoritma yang meminimumkan kebutuhan waktu dan ruang. Kompleksitas ruang, $S(n)$, diukur dari memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran masukan n .

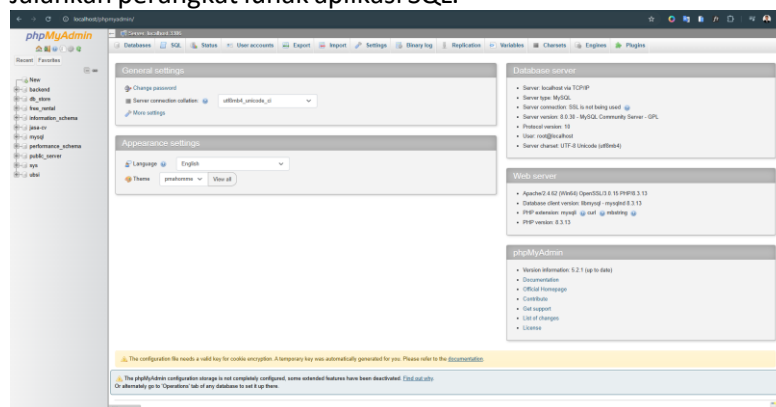
Unit Kompetensi	Kode Unit	:	J. 620100.020.02
	Judul Unit	:	Menggunakan SQL

LANGKAH KERJA

1. Persiapkan dan pasang perangkat lunak aplikasi data deskripsi/SQL.
 - 1.1. Pasang perangkat lunak aplikasi SQL sesuai dengan prosedur operasional standar.



- 1.2. Jalankan perangkat lunak aplikasi SQL.



2. Menggunakan fitur aplikasi SQL untuk membuat basis data
 - 2.1. Identifikasi fitur Fitur pengolahan DDL.
DDL (Data Definition Language) seperti CREATE, ALTER, dan DROP.
 - 2.2. Eksekusi fitur pengolahan DDL sesuai kebutuhan.
Database dibuat melalui laravel langsung saat pembuatan aplikasi dengan menerapkan MVC (model View Controler) dan migrasi tabel
3. Mengisi tabel
 - 3.1. Isi data tabel menggunakan perintah DML.

```
INSERT INTO level (nama_level) VALUES ('Admin'), ('Petugas'), ('Agen');
INSERT INTO tarif (daya, tarifperkwh) VALUES ('450VA', 1000), ('900VA', 1500),
('1300VA', 2000), ('2200VA', 2500);
```

3.2. Buat indeks.

```
CREATE INDEX idx_penggunaan_bulan_tahun ON penggunaan (bulan, tahun);
CREATE INDEX idx_tagihan_bulan_tahun ON tagihan (bulan, tahun);
```

3.3. Buatlah view tabel sesuai kebutuhan.

```
CREATE VIEW v_penggunaan AS
SELECT p.nama_pelanggan, u.bulan, u.tahun, u.meter_awal, u.meter_akhir
FROM penggunaan u
JOIN pelanggan p ON u.id_pelanggan = p.id_pelanggan;
```

4. Melakukan operasi relasional

4.1. Identifikasi fitur pengolahan DML

DML (Data Manipulation Language) seperti *INSERT*, *UPDATE*, *DELETE*, dan *SELECT*.

4.2. Gunakan perintah DML untuk manipulasi antar tabel.

Menambahkan data baru, mengubah data yang ada, dan menghapus data.

4.3. Gunakan Perintah DML untuk manipulasi antar-view.

Melakukan query pada view yang sudah dibuat.

4.4. Tulis perintah DML secara efisien.

Optimalkan query untuk performa yang lebih baik.

5. Membuat *stored procedure*

5.1. Buat *stored Procedure* dengan perintah SQL.

```
DELIMITER //
CREATE PROCEDURE GetPelanggan900VA()
BEGIN
    SELECT * FROM pelanggan WHERE id_tarif IN (SELECT id_tarif FROM tarif WHERE
    daya = '900VA');
END //
DELIMITER ;
```

5.2. Uji dan periksa prosedur *input* dan *outputnya*.

```
CALL GetPelanggan900VA();
```

6. Membuat *function*

6.1. Buat *function* dibuat dengan perintah SQL.

```
DELIMITER //
CREATE FUNCTION TotalPenggunaanPerBulan(pelanggan_id INT, bulan
VARCHAR(20), tahun YEAR)
RETURNS INT
BEGIN
    DECLARE total_penggunaan INT;
    SELECT SUM(meter_akhir - meter_awal) INTO total_penggunaan
    FROM penggunaan
    WHERE id_pelanggan = pelanggan_id AND bulan = bulan AND tahun = tahun;
    RETURN total_penggunaan;
END //
DELIMITER ;
```

6.2. Tulis perintah SQL pada *function* secara efisien.

```
DELIMITER //
CREATE TRIGGER after_penggunaan_insert
AFTER INSERT ON penggunaan
FOR EACH ROW
```



```

BEGIN
    DECLARE total_meter INT;
    SET total_meter = NEW.meter_akhir - NEW.meter_awal;
    INSERT INTO tagihan (id_penggunaan, id_pelanggan, bulan, tahun, jumlah_meter,
status)
    VALUES (NEW.id_penggunaan, NEW.id_pelanggan, NEW.bulan, NEW.tahun,
total_meter, 0);
END //
DELIMITER ;

```

7. Membuat trigger

7.1. Buat *trigger* dengan perintah SQL.

```

DELIMITER //
CREATE TRIGGER after_penggunaan_insert
AFTER INSERT ON penggunaan
FOR EACH ROW
BEGIN
    DECLARE total_meter INT;
    SET total_meter = NEW.meter_akhir - NEW.meter_awal;
    INSERT INTO tagihan (id_penggunaan, id_pelanggan, bulan, tahun, jumlah_meter,
status)
    VALUES (NEW.id_penggunaan, NEW.id_pelanggan, NEW.bulan, NEW.tahun,
total_meter, 0);
END //
DELIMITER ;

```

7.2. Uji kesesuaian hasil *trigger*.

SESUAI

8. Melakukan perintah commit dan rollback

8.1. Lakukan perubahan data dengan perintah commit.

```

START TRANSACTION;
INSERT INTO tarif (daya, tarifperkwh) VALUES ('450VA', 1000), ('900VA', 1500);
COMMIT;

```

8.2. Lakukan pembatalan penulisan data dilakukan dengan *rollback*.

```

START TRANSACTION;
DELETE FROM pelanggan WHERE id_pelanggan = 1;
ROLLBACK;

```

Unit Kompetensi	Kode Unit	:	J. 620100.021.02
	Judul Unit	:	Menerapkan Akses Basis Data

LANGKAH KERJA

Aplikasi telah dibuat menggunakan framework PHP yaitu CodeIgniter, demo CRUD akan diterapkan di aplikasi tersebut

Unit Kompetensi	Kode Unit	:	J. 620100.023.02
	Judul Unit	:	Membuat Dokumen Kode Program

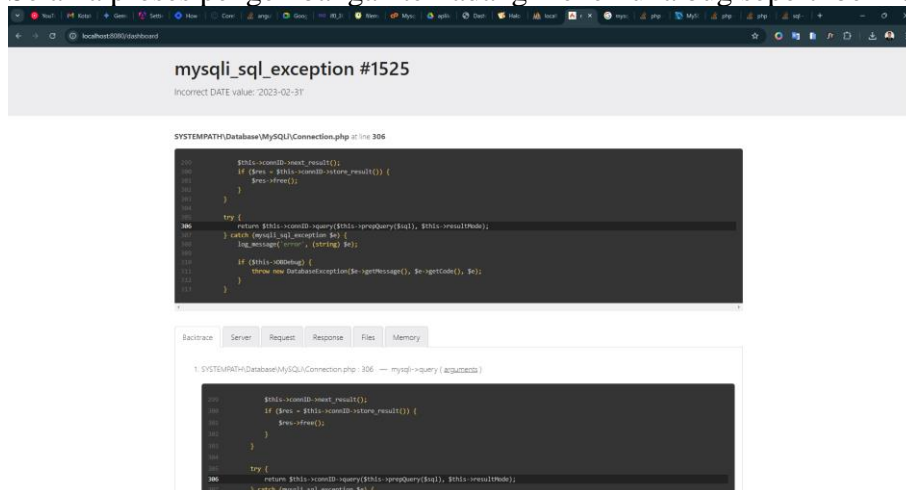
LANGKAH KERJA

Dokumen Kode Program ada di file program aplikasi Pencatatan (HaloNotes!)

Unit Kompetensi	Kode Unit	:	J.620100.025.02
	Judul Unit	:	Melakukan Debugging

LANGKAH KERJA

Selama proses pengembangan terkadang menemukan bug seperti berikut



Jadi saya mencoba untuk mencari solusi guna menangani bug tersebut.

Untuk mengatasi bug tersebut, saya memperbaiki format tipe data yang ada pada database dan aplikasi, disamakan.

Unit Kompetensi	Kode Unit	:	J.620100.031.001
	Judul Unit	:	Melakukan Profiling Program

LANGKAH KERJA

Ketika melakukan pengembangan aplikasi juga saya selalu memantau kinerja suatu aplikasi agar berjalan secara optimal. Saya menggunakan debugbar bawaan dari codeigniter 4, adapun kodenya sebagai berikut :

/*

* -----

* Debug Toolbar Listeners.

* -----

* If you delete, they will no longer be collected.

*/

```
if (CI_DEBUG && ! is_cli()) {  
    Events::on('DBQuery', 'CodeIgniter\\Debug\\Toolbar\\Collectors\\Database::collect');  
    Services::toolbar()->respond();  
  
    // Hot Reload route - for framework use on the hot reloader.  
    if (ENVIRONMENT === 'development') {  
        Services::routes()->get('__hot-reload', static function () {  
            (new HotReloader())->run();  
        });  
    }  
}
```

Unit Kompetensi	Kode Unit	:	J.620100.032.001
	Judul Unit	:	Melakukan <i>Code Review</i>

LANGKAH KERJA

Code Review dapat dilakukan di aplikasi Pencatatan (HaloNotes!)

Unit Kompetensi	Kode Unit	:	J.62010.033.02
	Judul Unit	:	Melaksanakan Pengujian Unit Program

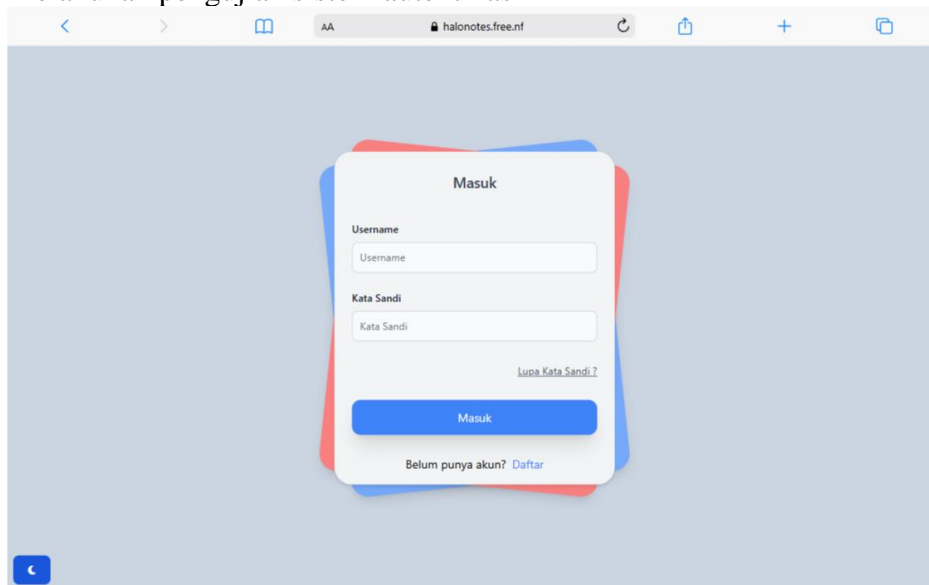
LANGKAH KERJA

1. Tentukan kebutuhan uji cobadalam pengembangan
 - a. Identifikasikan prosedur uji coba aplikasi sesuai dengan *software development lifecycle*.
Jawab:
 - Perencanaan Uji Coba: Tentukan skenario uji coba berdasarkan fungsi yang akan diuji.
 - Desain Uji Coba: Identifikasi unit-unit yang perlu diuji dan tentukan alat serta metode pengujian.
 - Implementasi Uji Coba: Tulis skrip uji unit untuk setiap fungsi menggunakan alat pengujian yang sesuai.
 - Eksekusi Uji Coba: Jalankan skrip uji unit dan catat hasilnya.
 - Evaluasi Uji Coba: Tinjau hasil pengujian dan identifikasi kesalahan atau area untuk perbaikan.
 - Perbaikan dan Pengulangan: Perbaiki kesalahan yang ditemukan dan jalankan kembali pengujian untuk memastikan perbaikan berhasil.
 - b. Tentukan tools uji coba.
PHPUnit: Alat pengujian unit standar untuk aplikasi Codeigniter.
 - c. Identifikasi standar dan kondisi uji coba.
Standar Uji Coba
 - Setiap fungsi harus memiliki cakupan pengujian minimal 80%.
 - Pengujian harus mencakup semua kondisi normal dan kasus batas.
 - Hasil uji harus didokumentasikan dengan baik.
 Kondisi Uji Coba
 - Uji coba dilakukan di lingkungan pengembangan yang mereplikasi produksi.
 - Data uji harus mencakup berbagai kondisi untuk menguji respons sistem terhadap input yang berbeda.
2. Persiapkandokumentasi uji coba
 - b. Tentukan kebutuhan untuk uji coba ditentukan.
 - Skrip uji untuk setiap fungsi utama aplikasi.
 - Data uji yang mencakup skenario normal dan batas.
 - c. Laksanakan Uji coba dengan variasi kondisi.
Dilakukan di terminal atau console dalam laravel php artisan
 - d. Buat skenario uji coba.
 - Menguji pembuatan catatan penggunaan dana baru.
 - Menguji perhitungan total piutang dalam yang belum terbayarkan.
 - Menguji penyimpanan data penggunaan dana.
3. Persiapkan data uji
 - a. Identifikasi data uji unit tes.
Data Hutang, Data Penggunaan Dana, Data Hajatan
 - b. Bangkitkan data uji unit tes.

4. Laksanakan prosedur uji coba
 - a. Desain skenario uji coba.
 - Buat catatan baru.
 - Catat debitur atau kreditur.
 - Buat rangkuman total hutang piutang.
 - Simpan data catatan.
 - b. Desain prosedur uji coba dalam algoritma.
Kode berada di codeigniter
 - c. Laksanakan uji coba.
Uji Coba di lakukan dengan aplikasi langsung tidak melalui config tes
5. Evaluasi hasil ujicoba
 - a. Catat hasil uji coba.
 - b. Analisis hasil uji coba.
 - c. Laporkan prosedur uji coba.
 - d. Selesaikan kesalahan/error

Unit Kompetensi	Kode Unit	:	J.62010.034.02
	Judul Unit	:	Melakukan Pengujian Integrasi Program

1. Melakukan pengujian sistem autentikasi



```

class Login extends BaseController
{
    public function login()
    {
        $rules = [
            'username' => [
                'rules' => 'required|alpha_dash',
                'errors' => [
                    'required' => 'Username harus diisi!',
                    'alpha_dash' => 'Username tidak mengandung spasi!'
                ]
            ],
            'password' => [
                'rules' => 'required|trim',
                'errors' => [
                    'required' => 'Kata sandi harus diisi!'
                ]
            ]
        ];

        if (!$this->validate($rules)) {
            $validasi = \Config\Services::validation();
            $errorInput = [
                'username' => $validasi->getError('username'),
                'password' => $validasi->getError('password')
            ];
            session()->setFlashdata('errorInput', $errorInput);
            return redirect()->to('/masuk')->withInput();
        }

        $username = htmlspecialchars($this->request->getPost('username'));
        $password = htmlspecialchars($this->request->getPost('password'));
    }
}

```

2. Melakukan salah satu penambahan catatan baru.

Tambah Data Piutang

Debitur: Test Debitur

Nominal: Rp. 500.000

Tanggal Piutang: 05/01/2025

Batal Tambah

Nama Debitur	Piutang	Status	Aksi
Garda Kalim Permadi	11	Belum Lunas	✓
Empluk Nashiruddin	14	Belum Lunas	✓
Elsa Aprilia	14	Belum Lunas	✓
Jamalia Uysi Kuswani	14	Belum Lunas	✓
Cahyo Latupono	19	Belum Lunas	✓
Yance Latika Handayani	25	Belum Lunas	✓
Warsita Arispatra Narpati S.Pd	Rp. 220.000	2023-08-16	Belum Lunas ✓
Virman Simanjuntak	Rp. 35.000	2023-08-02	Belum Lunas ✓
Karya Samosir	Rp. 35.000	2023-07-28	Belum Lunas ✓
Warji Nainggolan S.Farm	Rp. 50.000	2023-07-26	Belum Lunas ✓

```
class Piutang extends BaseController
{
    public function tambahDataPiutang()
    {
        $data = [
            'id' => rand(10000000, 99999999),
            'nama' => $this->request->getPost('debitur'),
            'nominal' => preg_replace("/[^\d-]/", "", $this->request->getPost('nominal')),
            'jenis' => 'Piutang',
            'status' => 'Belum Lunas',
            'tanggal' => $this->request->getPost('tanggal'),
            'id_user' => $this->user['id'],
            'created_at' => Time::now('Asia/Jakarta')
        ];

        $this->hutangData->insert($data);
        session()->setFlashdata('tambahPiutang', 'Piutang telah ditambahkan!');
        return redirect()->to('/piutang');
    }

    public function riwayatPiutang()
    {
        $data = [
            'title' => "Halo Piutang!",
            'heading' => "Data Piutang " . $this->user['nama'],
            'user' => $this->user,
            'piutangLunas' => $this->hutangData->where(['id_user' => $this->user['id'], 'status' => 'Lunas', 'jenis' => 'Piutang'])
        ];

        return view('riwayatPiutang', $data);
    }
}
```

3. Data akan tersimpan ke dalam database

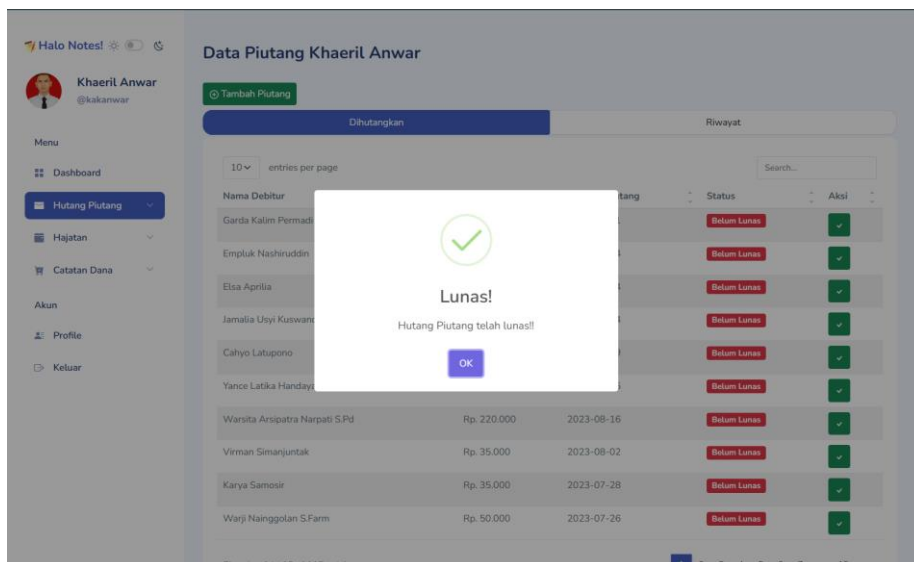
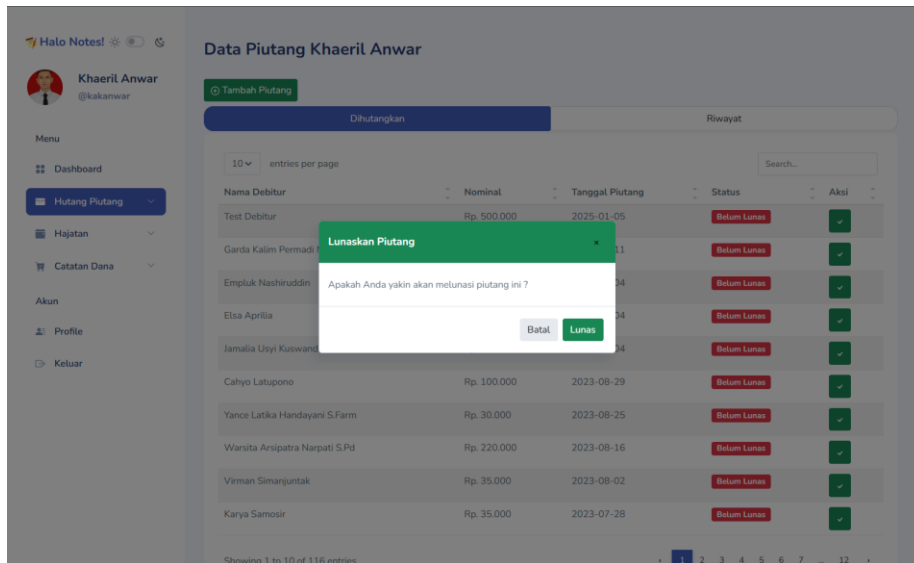
Berhasil!

Piutang telah ditambahkan!

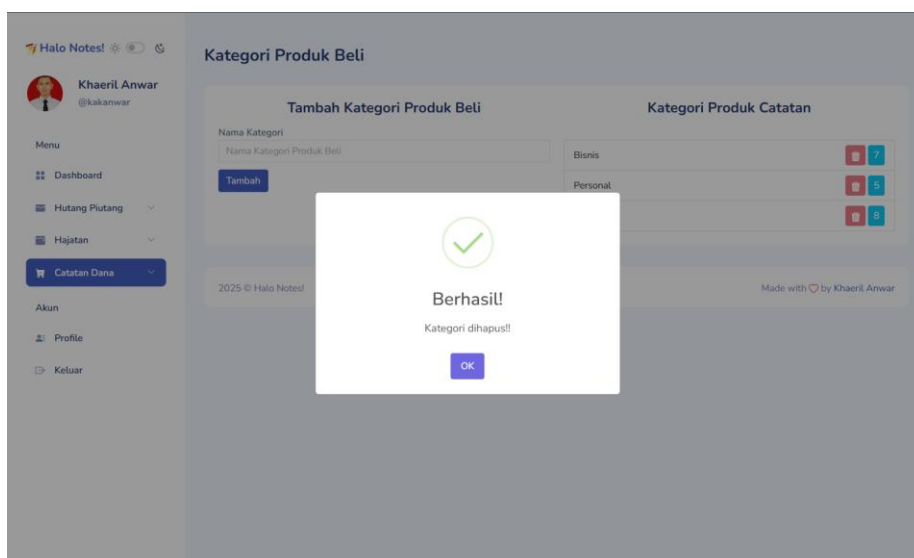
OK

Nama Debitur	Piutang	Status	Aksi
Test Debitur		Belum Lunas	✓
Garda Kalim Permadi		Belum Lunas	✓
Empluk Nashiruddin		Belum Lunas	✓
Elsa Aprilia		Belum Lunas	✓
Jamalia Uysi Kuswani		Belum Lunas	✓
Cahyo Latupono		Belum Lunas	✓
Yance Latika Handayani S.Farm	Rp. 30.000	2023-08-25	Belum Lunas ✓
Warsita Arispatra Narpati S.Pd	Rp. 220.000	2023-08-16	Belum Lunas ✓
Virman Simanjuntak	Rp. 35.000	2023-08-02	Belum Lunas ✓
Karya Samosir	Rp. 35.000	2023-07-28	Belum Lunas ✓

4. Mengubah status dari belum lunas menjadi lunas



5. Menghapus sebuah data, contohnya menghapus kategori catatan penggunaan dana.



```
7 class KategoriProdukBeli extends BaseController
36
37 public function storeKategori()
38 {
39     $rules = [
40         'nama_kategori' => [
41             'rules' => 'required|min_length[4]|trim',
42             'errors' => [
43                 'required' => 'Nama kategori tidak boleh kosong',
44                 'min_length' => 'Nama kategori minimal 4 karakter'
45             ]
46         ];
47     if (!$this->validate($rules)) {
48         return redirect()->back()->withInput();
49     }
50
51     $data = [
52         'id_user' => intval($this->user['id']),
53         'nama_kategori' => ucwords(strtolower($this->request->getPost('nama_kategori')))
54     ];
55     // return dd($data);
56     $this->kategoriModel->insert($data);
57     return redirect()->to('/kategori')->with('success', 'Kategori ditambahkan!');
58 }
59
60 public function hapus($id)
61 {
62     $this->kategoriModel->delete($id);
63     return redirect()->to('/kategori')->with('success', 'Kategori dihapus!');
64 }
```