



**PENGEMBANGAN APLIKASI *DASHBOARD* BERBASIS *WEB* UNTUK  
*MONITORING* PENJUALAN PRODUK *VENDING MACHINE* KATEGORI  
AIR**



**KERJA PRAKTIK**

**Program Studi  
S1 Sistem Informasi**

**UNIVERSITAS  
Dinamika**

**Oleh:**

**M. ILHAM RIDHO MANSYZ**

**21410100018**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2024**

**PENGEMBANGAN APLIKASI *DASHBOARD* BERBASIS *WEB* UNTUK  
*MONITORING* PENJUALAN PRODUK *VENDING MACHINE*  
KATEGORI AIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana



UNIVERSITAS  
**Dinamika**

**Disusun Oleh:**

**Nama : M. Ilham Ridho Mansyz**

**Nim : 21410100018**

**Program : S1 (Strata Satu)**

**Jurusan : Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

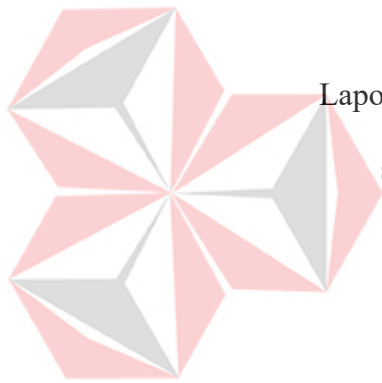
**2024**



“Kesuksesan tanpa kesulitan tidak berarti”

-M. Ilham Ridho Mansyz-

UNIVERSITAS  
Dinamika



Laporan Kerja Praktik ini saya persembahkan sepenuhnya dan saya dedikasikan untuk keluarga saya tercinta, dosen pembimbing saya, dan seluruh teman teman saya

UNIVERSITAS  
Dinamika

**LEMBAR PENGESAHAN**

**Pengembangan Aplikasi *Dashboard* Berbasis *Web* untuk *Monitoring*  
Penjualan Produk *Vending Machine* Kategori Air**

Laporan Kerja Praktik oleh

**M. Ilham Ridho Manysz**

NIM : 21410100018

Telah diperiksa, diuji, dan disetujui

Surabaya, 1 Juli 2024



**Agus Dwi Churniawan, S.Si., M.Kom**

NIDN. 0723088002

Disetujui

Penyelia,

**Jane Auditva**

Mengetahui,

Ketua Program Studi S1 Sistem Informasi

Digitally signed by   
Julianto  
Date: 2025.01.09  
16:11:04 +07'00'

**Julianto Lemantara, S.Kom., M.Eng.**

NIDN. 0722108601

**PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **M. Ilham Ridho Mansyz**  
NIM : **21410100018**  
Program Studi : **SI Sistem Informasi**  
Fakultas : **Fakultas Teknologi dan Informatika**  
Jenis Karya : **Laporan Kerja Praktik**  
Judul Karya : **PENGEMBANGAN APLIKASI *DASHBOARD* BERBASIS *WEB* UNTUK *MONITORING* PENJUALAN PRODUK *VENDING MACHINE* KATEGORI AIR**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 17 Januari 2025



**M. Ilham Ridho Mansyz**  
NIM : 21410100018

## ABSTRAK

Kerja praktik ini membahas pengembangan aplikasi *dashboard* berbasis *web* untuk memantau penjualan produk *vending machine* kategori air. Tujuan utama dari pengembangan aplikasi ini adalah untuk mempermudah proses analisis kinerja penjualan produk, sehingga pihak manajemen dapat membuat keputusan yang lebih tepat berdasarkan *data* yang tersedia. Aplikasi ini dibangun menggunakan teknologi web modern yang memungkinkan akses *data* dan menampilkan Informasi secara visual yang mudah dipahami. Melalui penerapan sistem ini, diharapkan dapat meningkatkan efisiensi operasional dan efektivitas strategi pemasaran dari produk *vending machine* yang diteliti.

**Kata Kunci:** Aplikasi *dashboard*, *Vending machine*, Visualisasi *data*, *Website*.



UNIVERSITAS  
Dinamika

## KATA PENGANTAR

Pertama-tama saya ucapkan puji dan syukur kehadirat Allah Subhanahu wa Ta'ala, yang telah memberikan Karunia-Nya sehingga Penulis dapat menyelesaikan laporan Kerja Praktik yang berjudul “Pengembangan *Dashboard* Berbasis Web untuk *Monitoring* Penjualan Produk *Vending Machine* Berkategori Air” dengan lancar.

Laporan kerja praktik ini digunakan sebagai syarat kelulusan dalam menyelesaikan program Sastra 1 Sistem Informasi Universitas Dinamika. Penyelesaian laporan Proyek Akhir ini tidak terlepas dari bantuan berbagai pihak eksternal maupun *internal* yang telah memberikan banyak masukan, nasihat, kritik dan saran, maupun dukungan kepada Penulis. Oleh karena itu, Penulis menyampaikan rasa terima kasih kepada :

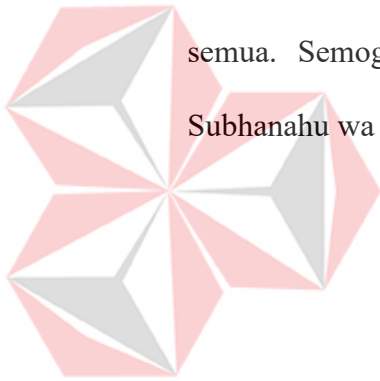
1. Orang tua yang selalu memotivasi, mendukung, serta mendoakan sehingga laporan kerja praktik daa.
2. Bapak Agus Dwi Churniawan, S.Si., M.Kom, selaku pembimbing yang selalu mendorong saya menjadi yang terbaik, memperbaiki kesalahan dan memberikan tantangan yang lebih sulit sehingga saya lebih termotivasi untuk membuat laporan dengan lebih baik.
3. Kepada kakak-kakak mentor yang terbaik yaitu Kak Nindhi dan Kak Najib yang selalu membimbing kami dalam proses berjalannya pembelajaran sehingga dapat menyerap ilmu dengan baik.
4. Pihak PT Revolusi Cipta Edukasi atau RevoU yang telah memberikan kesempatan dalam mengikuti program kampus merdeka, terima kasih juga atas



ilmu yang diberikan sehingga dapat saya terapkan dalam membuat aplikasi tugas praktik ini.

5. Seluruh pihak dan teman-teman yang tidak bisa saya sebutkan satu persatu, saya ucapkan terima kasih atas masukan dan saran sehingga kerja praktik dapat berjalan dengan lancar.

Akhir saya mengucapkan terima sekali lagi rasa terima kasih terhadap orangtua, bapak/ibu dosen, teman-teman, pihak RevoU, dan seluruh pihak yang telah memberikan saran, arahan, kritik, dan juga bimbingan sehingga seluruh kegiatan kerja praktik dapat berjalan dengan lancar. Saran, dan kritik akan selalu terbuka demi perbaikan projek maupun laporan sehingga dapat bermanfaat bagi kita semua. Semoga amal kebaikan yang bapak/ibu berikan dibalas oleh Allah Subhanahu wa Ta'ala. Sekian dari saya terima kasih.



UNIVERSITAS  
**Dinamika**

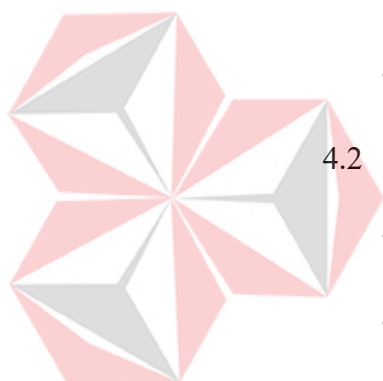
Surabaya, Agustus 2024

Penulis

## DAFTAR ISI

	<b>Halaman</b>
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
BAB II GAMBARAN UMUM PERUSAHAAN.....	4
2.1 Latar Belakang Perusahaan.....	4
2.2 Identitas Perusahaan.....	5
2.3 Visi Perusahaan.....	5
2.4 Misi Perusahaan .....	5
2.5 Struktur Organisasi .....	6
BAB III LANDASAN TEORI.....	8
3.1 <i>System Development Life Cycle (SDLC)</i> .....	8
3.2 <i>Waterfall</i> .....	10
3.3 <i>Hyper Text Markup Language (HTML)</i> .....	11

3.4	<i>Casecading Style Sheet (CSS)</i> .....	12
3.5	<i>Javascript</i> .....	12
3.6	<i>JavaScript Object Notation (JSON)</i> .....	12
3.7	<i>Blackbox</i> .....	14
3.8	<i>Vending machine</i> .....	14
3.9	<i>Website</i> .....	14
3.10	<i>Dashboard</i> .....	15
BAB IV DESKRIPSI PEKERJAAN .....		16
4.1	<i>Analysis</i> .....	16
4.1.1	Identifikasi Masalah .....	16
4.1.2	Analisis Kebutuhan .....	16
4.2	<i>Design</i> .....	19
4.2.1	Alur sistem .....	20
4.2.2	<i>Context diagram</i> .....	23
4.2.3	<i>Data flow</i> .....	23
4.2.4	Pemodelan <i>data</i> .....	25
4.3	<i>Implementation</i> .....	27
4.3.1	Fungsi dashboard (Fgs-01) .....	27
4.3.2	Fungsi filter (Fgs-02) .....	30
4.3.3	Fungsi tabel (Fgs-03) .....	30
4.4	<i>Testing</i> .....	31
4.5	<i>Maintenance</i> .....	36
BAB V PENUTUP .....		37



5.1 Kesimpulan .....	37
5.2 Saran.....	37
DAFTAR PUSTAKA .....	38
LAMPIRAN.....	39



UNIVERSITAS  
**Dinamika**

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 4. 1 Kebutuhan fungsional .....	17
Tabel 4. 2 Kebutuhan perangkat .....	19
Tabel 4. 3 Tabel hasil <i>test</i> aplikasi .....	32



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2. 1 Logo RevoU .....	4
Gambar 2. 2 Struktur PT. Revolusi Cita Edukasi .....	6
Gambar 3. 1 Alur pengembangan SDLC .....	8
Gambar 3. 2 Alur SDLC <i>model waterfall</i> .....	10
Gambar 3. 3 Struktur Objek JSON .....	13
Gambar 4. 1 Alur sistem fungsi <i>dashboard</i> (Fgs-01) .....	20
Gambar 4. 2 Alur sistem fungsi <i>filter</i> (Fgs-01).....	21
Gambar 4. 3 Alur sistem fungsi tabel (Fgs-03).....	22
Gambar 4. 4 <i>Context diagram</i> .....	23
Gambar 4. 5 <i>Data flow</i> .....	23
Gambar 4. 6 Struktur <i>data</i> JSON .....	25
Gambar 4.7 <i>Dashboard</i> .....	27
Gambar 4. 8 Sub fungsi total penjualan per produk dan harga.....	28
Gambar 4. 9 Sub fungsi total penjualan setiap tipe pembayaran .....	28
Gambar 4. 10 Sub fungsi total penjualan perbulan .....	29
Gambar 4. 11 Sub fungsi total penjualan setiap lokasi .....	29
Gambar 4. 12 Total penjualn seluruh pendapatan dan penjualan .....	30
Gambar 4.13 Filter .....	30
Gambar 4.14 Tabel.....	30

## DAFTAR LAMPIRAN

	<b>Halaman</b>
Lampiran 1. Surat Balasan dari Perusahaan.....	39
Lampiran 2. Log Bulanan .....	40
Lampiran 3. Rencana Kegiatan.....	47
Lampiran 4. Sertifikat MSIB .....	48
Lampiran 5. Transkrip Nilai .....	49
Lampiran 6. Kartu Bimbingan Kerja Praktik.....	50
\Lampiran 7. Biodata Penulis.....	51



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

PT Revolusi Cita Edukasi, atau yang biasa dikenal dengan singkatannya yaitu RevoU adalah perusahaan rintisan (*startup*) di bidang pendidikan teknologi yang berlokasi di Jakarta, Indonesia. RevoU berfokus pada dunia pendidikan yang memberikan pelatihan intensif (*bootcamp*) mempersiapkan individu dalam menghadapi dunia kerja dalam bidang teknologi, yang berfokus pada sektor digital.

RevoU bekerja sama dengan program MBKM (Merdeka Belajar Kampus Merdeka) dari Kemendikbud sebagai mitra penyedia program Studi Independen Bersertifikat (MSIB). Melalui kerja sama ini, RevoU membuka kesempatan bagi mahasiswa dari berbagai perguruan tinggi di Indonesia untuk mengikuti program bootcamp intensif di bidang teknologi digital, seperti *Data Analytics*, *Product Management*, *Digital Marketing*, dan *Software Engineering*, sebagai bagian dari program MBKM.

Program bootcamp RevoU yang terintegrasi dengan MBKM memberikan pengalaman belajar yang intensif dan terstruktur selama beberapa minggu kepada mahasiswa, dengan fokus pada pengembangan keterampilan praktis yang dibutuhkan di industri. Setelah menyelesaikan program, mahasiswa akan mendapatkan sertifikat kompetensi yang diakui oleh industri dan dapat meningkatkan daya saing mereka di pasar kerja. Program ini juga dapat dikonversi menjadi SKS (Satuan Kredit Semester) di perguruan tinggi asal mahasiswa,



sehingga mereka dapat tetap melanjutkan studi formal mereka sambil mendapatkan pengalaman praktis di industri.

Salah satu program tersebut adalah *data analytics* dan *software development* dimana pada program tersebut memberikan materi kepada para peserta yang mengikuti program mengenai bagaimana cara memanfaatkan data dan teknologi untuk dimanfaatkan dalam kepentingan bisnis.

Kombinasi antara *data analytics* dan *software development* memiliki peran penting dalam mendorong pertumbuhan bisnis di era digital. *Data analytics* memungkinkan perusahaan untuk mengambil keputusan yang lebih baik, meningkatkan efisiensi operasional, memahami pelanggan dengan lebih baik, dan mengidentifikasi peluang pasar baru melalui pengumpulan, pengolahan, dan analisis data.

Aplikasi “Dashboard Berbasis Web untuk *Monitoring* Penjualan *Vending Machine* Kategori Air” dirancang untuk memudahkan tim penjualan dalam memantau penjualan produk. Melalui *dashboard* ini, tim penjualan dapat mengakses Informasi seperti jumlah penjualan per produk dan tipe pembayaran (kredit atau lunas). Sebelumnya, tim harus mengunjungi *vending machine* untuk mengetahui pendapatan, tetapi dengan integrasi *website*, Informasi ini kini dapat diakses dari mana saja dan kapan saja. Hal ini meningkatkan efisiensi dan memudahkan pengambilan keputusan berdasarkan *data* penjualan.

Akhir kata “Aplikasi *Dashboard* Berbasis *Web* untuk *Monitoring* Penjualan Produk *Vending Machine* Kategori Air” dapat menyelesaikan banyak permasalahan mulai dari pembuatan keputusan yang lebih objektif dari pada sekedar insting karena berdasarkan data, menghemat waktu dalam pengecekan

*vending machine* karena dapat mengetahui kinerjanya dimana saja, dan masih banyak lagi permasalahan yang bisa diselesaikan jika menggunakan aplikasi tersebut.

## 1.2 Rumusan Masalah

Bagaimana cara mengembangkan aplikasi *dashboard* yang sesuai dengan kebutuhan pengguna.

## 1.3 Batasan Masalah

1. Dasbor hanya spesifik menunjukkan data pada produk kategori air.
2. Dasbor memiliki 2 fungsi saja yaitu untuk menampilkan data, filter data, dan tidak memiliki fitur *CUD* pada data.

## 1.4 Tujuan

Berdasarkan permasalahan yang diuraikan di atas, maka tujuan dari kerja praktik ini adalah aplikasi untuk reservasi tempat olahraga berbasis Laravel dengan nama “SportSpotter.” Perancangan ini bertujuan untuk memberikan kemudahan.

## 1.5 Manfaat

Pembuatan aplikasi *dashboard* memiliki manfaat sebagai berikut:

1. Pemantauan kinerja *vending machine* dapat dilakukan dimana saja dan kapan saja.
2. Mendapatkan *insight* yang bisa digunakan untuk mencari permasalahan, menemukan solusi, serta mencari peluang bisnis.
3. Mendorong keputusan bisnis yang lebih *objektif* karena berdasarkan *data*.

## BAB II

### GAMBARAN UMUM PERUSAHAAN

#### 2.1 Latar Belakang Perusahaan



Gambar 2. 1 Logo RevoU

PT Revolusi Cita Edukasi (RevoU) adalah perusahaan rintisan teknologi pendidikan (edutech) yang didirikan pada tahun 2019 di Jakarta, Indonesia. Berangkat dari keprihatinan akan kesenjangan antara lulusan perguruan tinggi dan kebutuhan industri di era digital, RevoU hadir untuk menjembatani kesenjangan tersebut dengan menyediakan program pelatihan intensif (*bootcamp*) yang berfokus pada pengembangan keterampilan praktis di bidang teknologi. Logo RevoU bisa dilihat pada gambar diatas.

RevoU berkomitmen untuk memberikan pendidikan berkualitas tinggi yang relevan dengan kebutuhan industri, sehingga lulusannya dapat bersaing dan berkontribusi secara optimal di dunia kerja. Melalui pendekatan pembelajaran yang inovatif, kurikulum yang adaptif, dan instruktur yang berpengalaman di industri, RevoU telah berhasil mencetak banyak talenta digital yang siap menghadapi tantangan di era revolusi industri 4.0.

## 2.2 Identitas Perusahaan

Nama Instansi : Revolusi Cita Edukasi

Alamat : Go Work Menara Rajawali, Ground Floor, Jl. DR. Ide Anak Agung Gde Agung, RT.5/RW.2, Kuningan, Jakarta, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12950

No. Telepon : [+62813-9910-0086](tel:+62813-9910-0086)

Email : [admissions@revou.co](mailto:admissions@revou.co)

## 2.3 Visi Perusahaan

*Develop the best professional learning platform in Southeast Asia, to give millions of people access to better career opportunities*

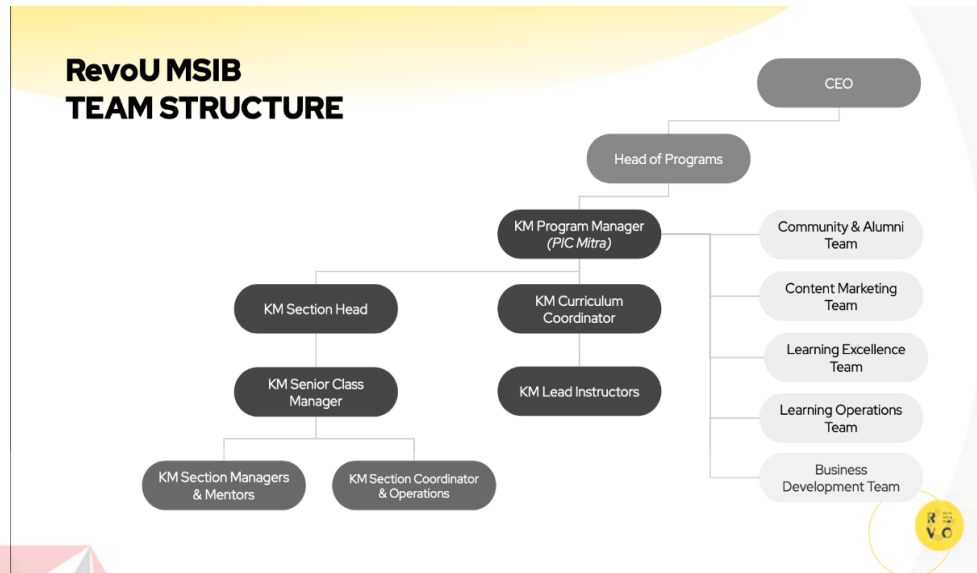
## 2.4 Misi Perusahaan

Misi dari PT Amanah Karya Indonesia adalah sebagai berikut:

1. *Kindess*
  - *Empathy*
  - *Respect*
  - *Selflessness*
2. *Candor*
  - *Authenticity*
  - *Feedback seeking*
  - *Outspoken*
  - *Failing safety\*
3. *Highest standards*
4. *Passion*

## 2.5 Struktur Organisasi

Berikut gambar struktur organisasi PT. Revolusi Cita Edukasi, dan perannya masing-masing yang bisa dilihat pada gambar dibawah ini:




Gambar 2. 2 Struktur PT. Revolusi Cita Edukasi

Struktur organisasi pada gambar menunjukkan hierarki tim RevoU MSIB.

Berikut adalah penjelasan dari setiap level dalam struktur tersebut:

1. **CEO (Chief Executive Officer):** Pimpinan tertinggi.
2. **Head of Programs:** Melapor kepada CEO, mengelola keseluruhan program.
3. **KM Program Manager (PIC Mitra):** Melapor kepada Head of Programs, mengelola program KM.
4. **KM Section Head:** Melapor kepada KM Program Manager, mengawasi manajer kelas senior.
5. **KM Senior Class Manager:** Melapor kepada KM Section Head, mengelola kelas senior.

6. **KM Section Managers & Mentors:** Melapor kepada *KM Senior Class Manager*, membimbing peserta.
7. **KM Section Coordinator & Operations:** Melapor kepada *KM Senior Class Manager*, mengelola koordinasi seksi dan operasi.
8. **KM Curriculum Coordinator:** Melapor kepada *Head of Programs*, mengembangkan kurikulum.
9. **KM Lead Instructors:** Melapor kepada *Head of Programs*, memimpin instruktur.
10. **Community & Alumni Team:** Mengelola komunitas dan alumni.
11. **Content Marketing Team:** Bertanggung jawab atas pemasaran konten.
12. **Learning Excellence Team:** Fokus pada peningkatan kualitas pembelajaran.
13. **Learning Operations Team:** Mengelola operasi pembelajaran.
14. **Business Development Team:** Bertanggung jawab atas pengembangan bisnis.



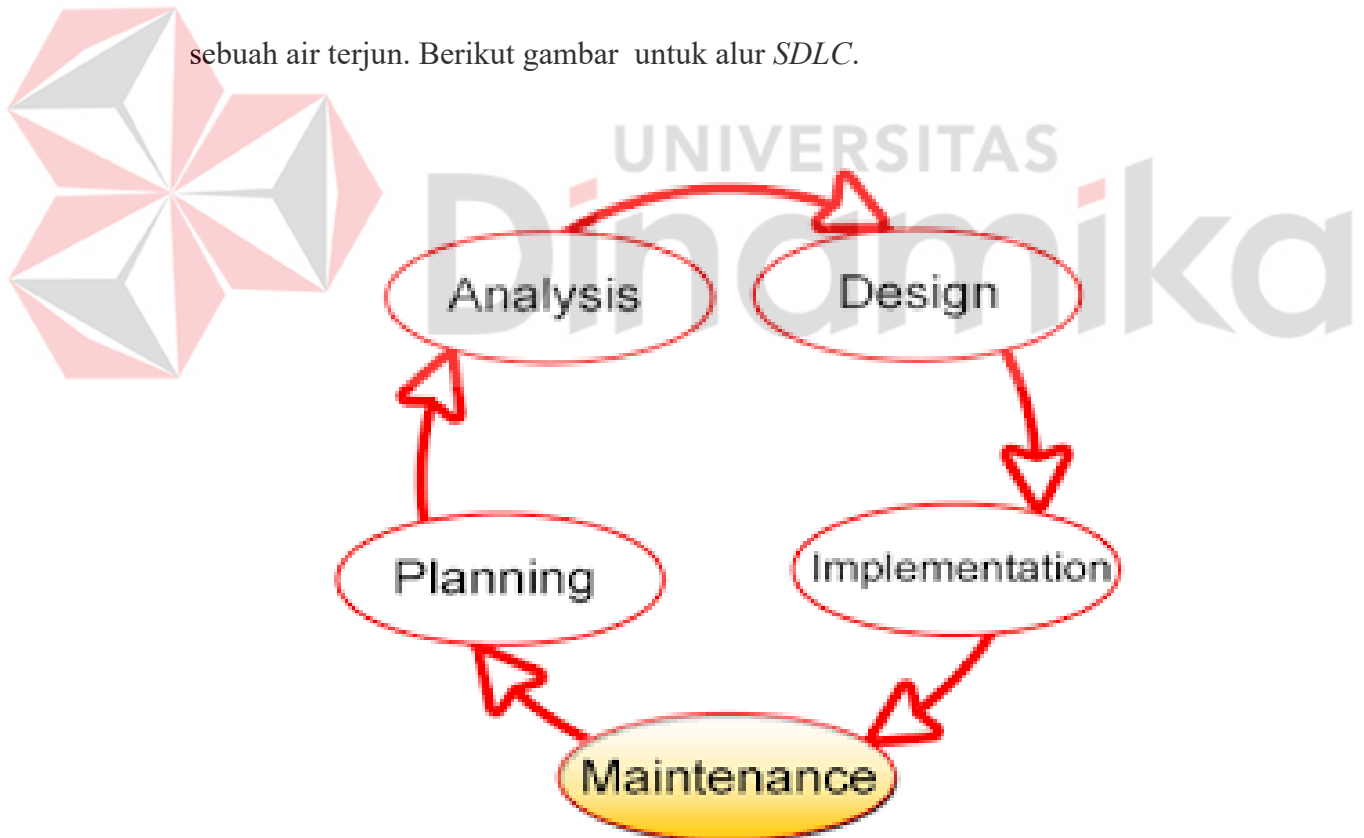
Struktur ini menunjukkan hubungan pelaporan dan tanggung jawab masing-masing tim dalam mendukung keberhasilan program RevoU MSIB.

## BAB III

### LANDASAN TEORI

#### 3.1 *System Development Life Cycle (SDLC)*

*SDLC* adalah suatu proses pengembangan *software*, mulai dari perencanaan, perancangan, hingga pengetesan sebuah *software* (Fitria Nur Hasanah et al., 2020). *SDLC* memastikan agar proses pembuatan *software* lebih terarah, transparan, dan efisiensi tetap terjaga. Terdapat banyak model *SDLC* yang bisa digunakan salah satu yang umum digunakan adalah model *waterfall* dimana pada model tersebut mengikuti proses yang lebih terstruktur dan mengalir layaknya sebuah air terjun. Berikut gambar untuk alur *SDLC*.



Gambar 3. 1 Alur pengembangan SDLC (Fitria Nur Hasanah et al., 2020)

Berikut penjelasan mengenai setiap alur *SDLC* :

1. *Planning*

Pada tahap ini mulai perencanaan proyek secara keseluruhan. Memulai dengan merencanakan dan memperhitungkan sumber daya, jadwal, dan anggaran.

2. *Analysis*

Tahap kedua adalah tahap analisis kebutuhan pengguna, apa saja yang diperlukan pengguna dalam aplikasi tersebut.

3. *Design*

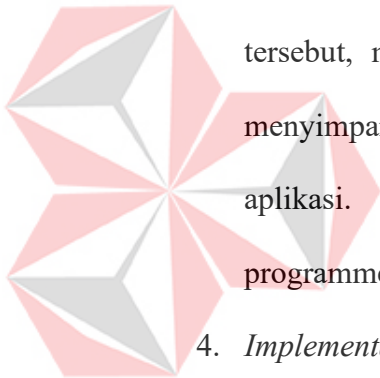
Tahapan *design* adalah tahapan bagaimana kita merancang aplikasi tersebut, mulai dari perancangan alur aplikasi, rancangan *database* untuk menyimpan data tersebut, rancangan aplikasi, dan rancangan alur data pada aplikasi. Rancangan tersebut memberikan pemahaman kepada para programmer untuk membuat aplikasi dengan jelas.

4. *Implementation*

Setelah proses pengodingan selesai sesuai rancangan, maka seluruh kode tersebut diintegrasikan sehingga menjadi sebuah satu kesatuan aplikasi.

5. *Testing*

Setelah aplikasi sudah terselesaikan maka proses selanjutnya, adalah proses pengujian aplikasi. Proses tersebut akan menguji setiap fungsi yang ada di dalam aplikasi apakah *data* muncul, apakah *data* tersimpan, dll.





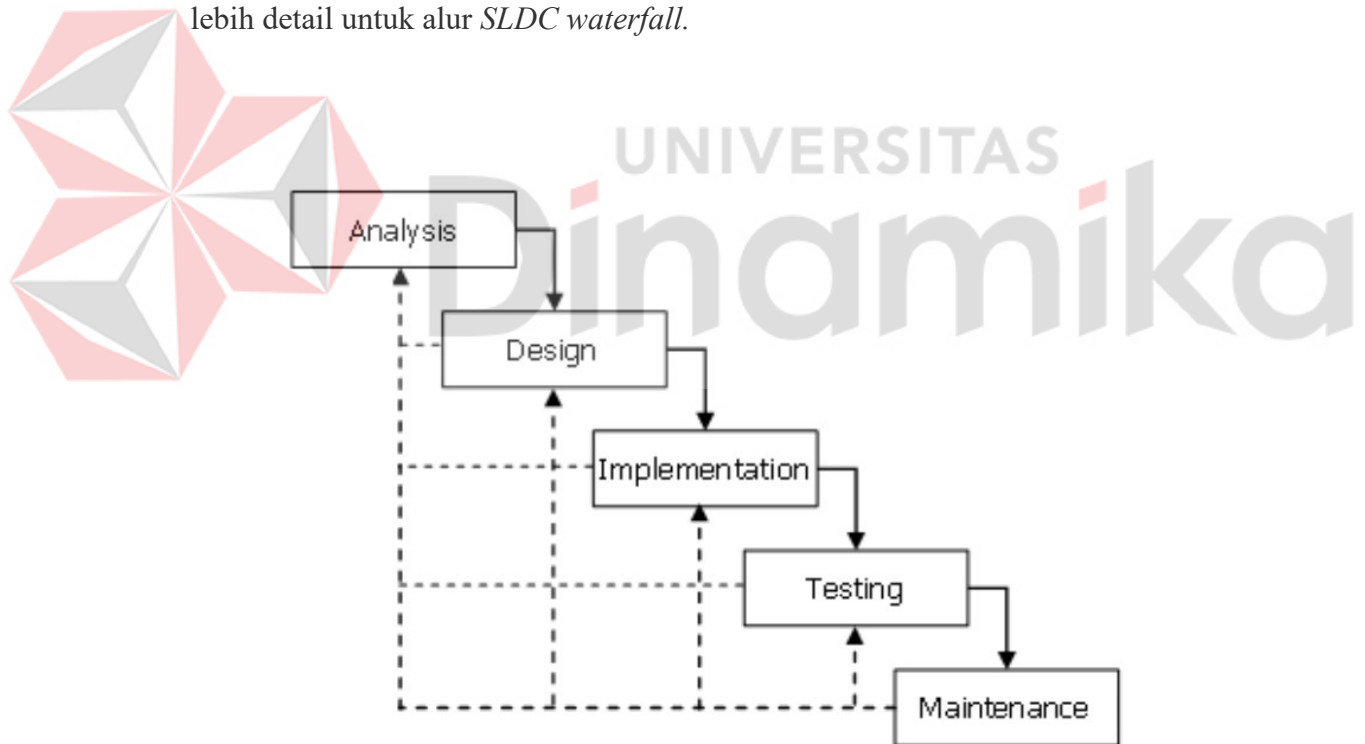
## 6. *Maintenance*

Proses terakhir adalah proses pemeliharaan aplikasi setelah aplikasi sudah dipastikan siap, maka aplikasi siap di rilis, namun tetap ada proses perbaikan bila ada kerusakan pada aplikasi tersebut setelah digunakan.

### 3.2 *Waterfall*

Model *waterfall* adalah salah satu model *SDLC* pertama yang digunakan untuk merancang sebuah aplikasi / *system* model tersebut mengedepankan Langkah yang sistematis dan terencana mulai dari analisis kebutuhan, desain, implementasi, *testing*, dan pemeliharaan (Syarif & Pratama, 2021). Berikut gambar menjelaskan

lebih detail untuk alur *SLDC waterfall*.



Gambar 3. 2 Alur SDLC *model waterfall* (Bassil, 2012)

Berikut penjelasan yang lebih jelas untuk masing masing proses :

1. *Requirement*

Langkah awal adalah dengan melakukan analisis kebutuhan mulai dari kebutuhan fungsional / fitur aplikasi, kebutuhan non fungsional yaitu kebutuhan yang menunjang aplikasi, dan kebutuhan perangkat untuk pembuatan aplikasi.

2. *Design*

Setelah analisis kebutuhan selanjutnya, dilakukan proses desain sistem, mulai dari desain alur data struktur data, dan model data. Desain tersebut akan digunakan untuk merancang aplikasi dan mengetahui alurnya,

3. *Implementation*

Implementasi adalah bagaimana tampilan aplikasi ditampilkan, Bagaimana *website* tersebut, dan halaman apa saja yang ada pada *website* tersebut.

4. *Testing*

Proses pengujian adalah proses apakah semua kebutuhan pengguna / fitur aplikasi yang dibutuhkan oleh pengguna sudah terpenuhi.

5. *Maintenance*

Terakhir adalah proses pemeliharaan sistem, yang akan dilakukan secara berkala untuk menjaga fungsionalitas aplikasi jika terdapat *error*

### 3.3 *Hyper Text Markup Language (HTML)*

Html adalah bahasa pemrograman yang umum menjadi kerangka dalam pembuatan *website*. Kerangka tersebut lah yang menjadi dasar sebuah *website* yang akan dimanipulasi dengan menggunakan *CSS* sehingga menjadi lebih ramah pengguna. Seluruh *website* akan membutuhkan html sebagai kerangka dasar

*website* mereka, kerangka dasar tersebut ialah yang akan dimanipulasi dan menjadi struktur awal sehingga dapat diberikan fungsionalitas dengan *CSS* dan *javascript* (Batalas et al., 2021).

### 3.4 *Cascading Style Sheet (CSS)*

*CSS* adalah bahasa pemrograman yang digunakan untuk mengatur komponen dalam web tepatnya mengatur bagian dari *html*. *CSS* memungkinkan kita memanipulasikan komponen dalam *html* sehingga dapat menampilkan sesuai dengan kemampuan kita, dengan *CSS* kita bisa mengatur tataletak, merubah *font*, dan mengatur ukuran *font*. Kemampuan memanipulasi tersebut menambah nilai aksesibilitas kita kepada pengguna baru sehingga dapat dengan mudah memahami *website* yang kita buat (Ika Ratna Indra Astutik & Mochamad Alfian Rosid, 2019).

### 3.5 *Javascript*

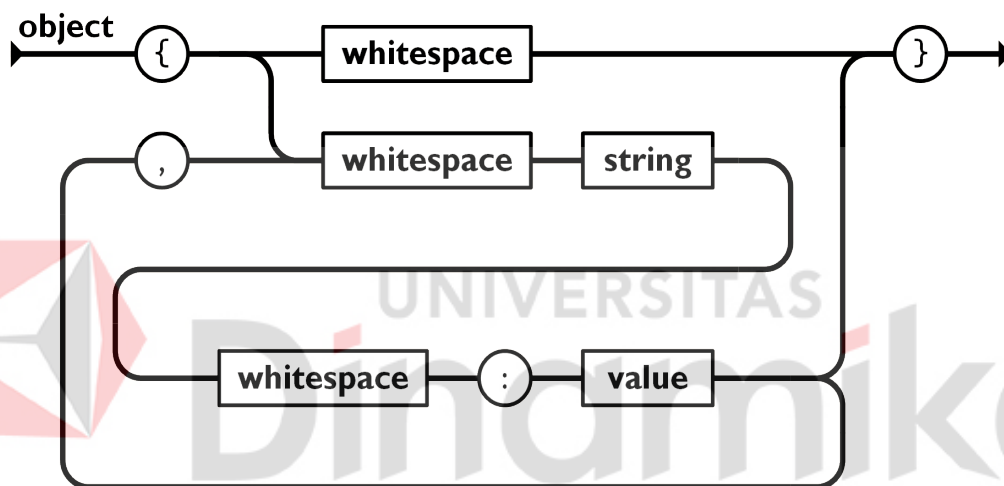
*Javascript* adalah skrip yang digunakan bersamaan dengan *Html* yang fungsi utamanya adalah untuk melakukan proses dari sisi pengguna, proses tersebut bisa berupa validasi *inputan form*, proses perhitungan, dan proses proses lainnya. *Javascript* berbeda dengan bahasa pemrograman *java* dimana *javascript* dijalankan di klien (*browser*) namun *java* dijalankan oleh aplikasi pemrograman itu sendiri (Wirfs-Brock & Eich, 2020).

### 3.6 *JavaScript Object Notation (JSON)*

*JSON* atau yang biasa dikenal dengan *JavaScript Object Notation data format* adalah bagian dari bahasa pemrograman *javascript*. Meskipun bagian dari *javascript* namun *Json* tidak memiliki fungsi yang seperti *javascript*. *JSON* adalah

bagian dari bahasa pemrograman yang biasanya digunakan sebagai *format* penyimpanan data antar perangkat (Bourhis et al., 2020).

Kegunaan yang bisa menyimpan *data* tersebut dan bisa digunakan oleh berbagai perangkat, maka biasanya sering terjadinya pertukaran *data* antara *browser* dengan *server*, dan *server* dengan *server*. Namun proses pertukaran tersebut bisa digunakan dalam berbagai situasi tidak hanya oleh *browser* dan *server*. Berikut gambar untuk memperjelasnya.



Gambar 3. 3 Struktur Objek JSON

JSON terdiri dari dua struktur utama: *objek* dan larik. Objek dalam JSON adalah kumpulan pasangan nama/nilai yang tidak terurut, ditandai dengan kurung kurawal { } dan setiap pasangan dipisahkan oleh koma, dengan nama dan nilai dipisahkan oleh titik dua. Larik adalah kumpulan nilai yang terurut, dimulai dengan tanda kurung siku [ ] dan setiap nilai dipisahkan oleh koma. Struktur ini memungkinkan JSON untuk menjadi alat yang fleksibel dan efisien dalam pertukaran *data* antarbahasa pemrograman yang berbeda, memudahkan proses pemrosesan dan integrasi *data* dalam aplikasi.

### 3.7 Blackbox

*Blackbox test* dijelaskan sebagai metode pengujian perangkat lunak yang fokus pada evaluasi fungsionalitas sistem tanpa memerlukan pengetahuan tentang struktur *internalnya* (Zaidir et al., 2023).

Metode ini digunakan untuk:

1. Menguji *Input* dan *Output*: Memastikan bahwa untuk setiap *input* yang diberikan, *output* yang dihasilkan sesuai dengan yang diharapkan.
2. Mendeteksi Cacat: Menemukan kesalahan pada fungsi sistem aplikasi dan menu aplikasi yang hilang.
3. *Equivalence Partitioning*: Teknik yang membagi *data input* menjadi beberapa partisi untuk mengurangi jumlah *test case* dan mengidentifikasi kelas kesalahan.

### 3.8 Vending machine

*Vending machine* adalah mesin yang berfungsi sebagai toko tanpa awak untuk kenyamanan manusia. Mesin penjual otomatis pertama kali ditemukan oleh Heron dari Alexandria dan telah berkembang sejak abad ke-19. Saat ini, *vending machine* berkembang dari toko tunggal fungsi menjadi stasiun yang dapat menyediakan berbagai layanan. *Vending machine* memiliki ukuran yang kompak sehingga dapat dipasang di lokasi yang tidak cocok untuk bisnis ritel lainnya (Cao et al., 2023).

### 3.9 Website

*Website* adalah halaman yang berisikan *file* Html (*Hypertext Markup Language*) dimana pada *file* Html tersebut berisikan instruksi yang dapat

memunculkan teks, gambar, audio, animasi, dan video yang bisa diakses melalui melalui *browser* seperti google chrome, microsoft edge, dll dimana dapat diakses oleh seluruh orang dengan menggunakan *internet* (Hasan & Abuelrub, 2011).

### 3.10 *Dashboard*

*Dashboard* adalah tampilan visual yang berisi Informasi penting guna mencapai tujuan. Informasi tersebut dapat berupa sebuah matriks, atau *key performance indicator* (KPI) yang berasal dari *data* kompleks kemudian diolah menjadi sebuah informasi dan di visualisasikan sehingga dapat diinformasikan kepada pengguna (Bach et al., 2023).

*Dashboard* menawarkan berbagai manfaat yang signifikan dalam manajemen dan pengambilan keputusan. Beberapa manfaat utama dari *dashboard* adalah:

1. **Visualisasi *Data*:** Menyajikan *data* dalam bentuk grafik dan tabel yang mudah dipahami.
2. **Pengambilan Keputusan:** Membantu membuat keputusan yang lebih baik dan berbasis *data*.
3. **Efisiensi Waktu:** Mengkonsolidasikan *data* dari berbagai sumber dalam satu tampilan.
4. **Peningkatan Komunikasi:** Memudahkan berbagi Informasi dengan tim atau pemangku kepentingan.
5. **Deteksi Masalah Dini:** Mengidentifikasi masalah atau risiko lebih awal.
6. **Pengukuran Kinerja:** Memantau pencapaian target dan tujuan strategis.
7. **Aksesibilitas:** Akses *data* dari berbagai perangkat kapan saja dan di mana saja.

## BAB IV

### DESKRIPSI PEKERJAAN

#### 4.1 *Analysis*

##### 4.1.1 Identifikasi Masalah

Hasil dari wawancara dan observasi bahwa tim sales memiliki beberapa masalah berikut daftar masalah tersebut:

1. Proses pemantauan yang tidak efisien dan tidak efektif karena harus mengecek satu per satu *vending machine*. Dimana lokasi mesin tersebut berada pada tempat yang berbeda-beda jika diperiksa / pantau satu persatu akan membutuhkan biaya yang lebih mahal dan waktu yang lama karena harus *mendatangi* mesin satu persatu.
2. Pengambilan Keputusan yang tidak objektif karena tidak berdasarkan *data*, dan hanya berdasarakan pengalaman atau asumsi pribadi. Pengambilan *data* yang objektif dapat membuat solusi dalam membuat rencana, membuat keputusan lebih tepat dan sesuai dengan kenyataan yang ada bukan berdasarkan pengalaman dan asumsi pribadi seseorang.

##### 4.1.2 Analisis Kebutuhan

Analisis kebutuhan membahas apa saja kebutuhan / fitur pada aplikasi yang dibutuhkan oleh pengguna untuk kegiatan bisnis mereka. Berdasarkan observasi dan wawancara, maka kebutuhan fungsional yang dibutuhkan adalah sebuah *filter* dan, *dashboard* yang dapat menampilkan sebuah visualisasi. Berikut tabel untuk penjelasan lebih detailnya.

Tabel 4. 1 Kebutuhan fungsional

Pengguna	Kode fungsi	Fungsi	Kode sub fungsi	Sub fungsi	Deskripsi	Batasan
User	Fgs-01	Dashboard	Fgs-dsb-01	Visualisasi total penjualan per produk dan harga.	Menampilkan <i>data</i> total penjualan setiap harga dan produknya.	Menampilkan total penjualan untuk setiap produk dan harganya saja.
			Fgs-dsb-02	Visualiasi total transaksi per tipe pembayaran	Menampilkan <i>data</i> total transaksi setiap tipe pembayaran	Menampilkan total penjualan untuk setiap tipe pembayarannya saja.
			Fgs-dsb-03	Visualisasi total penjualan perbulan	Menampilkan total penjualan setiap bulan	Menampilkan total penjualan setiap bulannya saja.
			Fgs-dsb-04	Visualiasi total penjualan perlokasi	Menampilkan total penjualan setiap lokasi	Menampilkan total penjualan setiap lokasinya saja.
			Fgs-dsb-05	<i>Scorecard</i> total penjualan secara keseluruhan	Menampilkan seluruh toal penjualan.	Menampilkan total penjualan saja.



Pengguna	Kode fungsi	Fungsi	Kode sub fungsi	Sub fungsi	Deskripsi	Batasan
			Fgs-dsb-06	<i>Scorecard</i> total pendapatan secara keseluruhan	Menampilkan seluruh total pendapatan.	Menampilkan total pendapatan saja.
	Fgs-02	<i>Filter</i>	Fgs-flt-01	<i>Filter</i> kuartal tahun	Pengguna dapat memilih <i>data</i> sesuai dengan kuartal tahun yang dipilih.	Pengguna hanya bisa memilih 1 jenis kuartal tahun atau semuanya, tidak bisa memilih 2 jenis kuartal.
Fgs-flt-02			<i>Filter</i> tipe pembayaran	Pengguna dapat memilih <i>data</i> sesuai dengan tipe pembayaran yang dipilih.	-	
Fgs-flt-03			<i>Filter</i> lokasi	Pengguna dapat memilih <i>data</i> sesuai dengan lokasi yang dipilih.	Pengguna hanya bisa memilih 1 jenis lokasi atau semuanya, tidak bisa memilih 2 lokasi.	
	Fgs-03	Tabel <i>data</i>	Fgs-tbl-01	<i>Search</i>	Memungkinkan pengguna mencari <i>data</i> yang diinginkan.	-

Setelah kita mengetahui apa saja kebutuhan pengguna pada aplikasi, selanjutnya dilakukan proses analisis kebutuhan non-fungsional, kebutuhan yang mendukung dijalanannya *website* tersebut.

Berikut kebutuhan non-fungsional yang akan dibutuhkan :

1. *Website* yang *responsive* untuk segala jenis perangkat mulai dari komputer laptop, *smartphone*, dan juga tabel.

Kebutuhan fungsional dan non-fungsional terpenuhi selanjutnya perlu identifikasi kebutuhan perangkat apa saja yang dibutuhkan dalam proses pembuatan aplikasi tersebut. Detail rinciannya bisa dilihat pada tabel dibawah ini.

Tabel 4. 2 Kebutuhan perangkat

Software	Hardware
1. <i>Visual Studio Code</i>	1. Jaringan Internet (kuota.wi-fi)
2. <i>Internet Browser</i>	2. <i>Hard disk</i> minimal 256 GB
3. GitHub	3. RAM minimal 4 GB
4. Vercel	

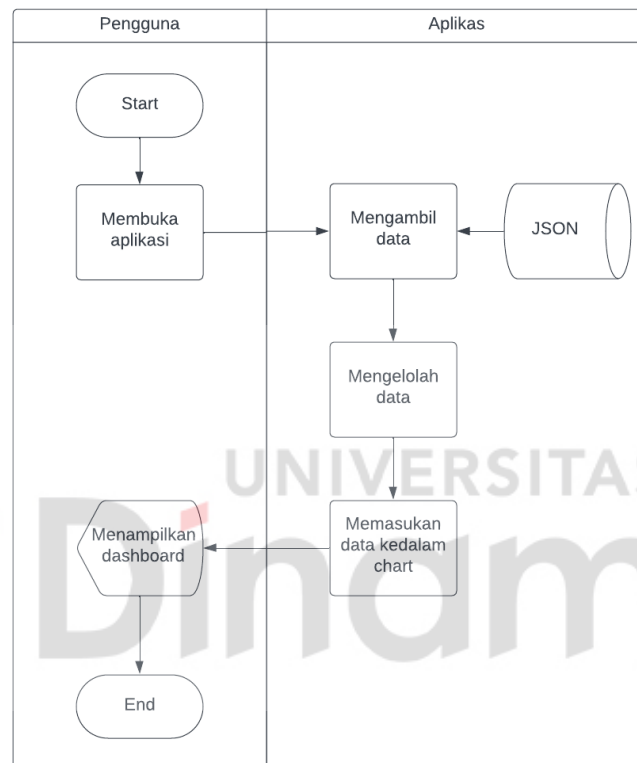
#### 4.2 Design

Pada tahap desain kita akan merancang bagaimana alur sistem tersebut bekerja. Mulai dari bagaimana *data* tersebut mengalir, bagaimana alur aplikasi, serta dari mana *data* tersebut berasal.

#### 4.2.1 Alur sistem

Pada bagian ini akan menjelaskan bagaimana alur penggunaan aplikasi mulai dari membuka *dashboard*, mengoprasikan tabel *data*, dan melakukan *filter*.

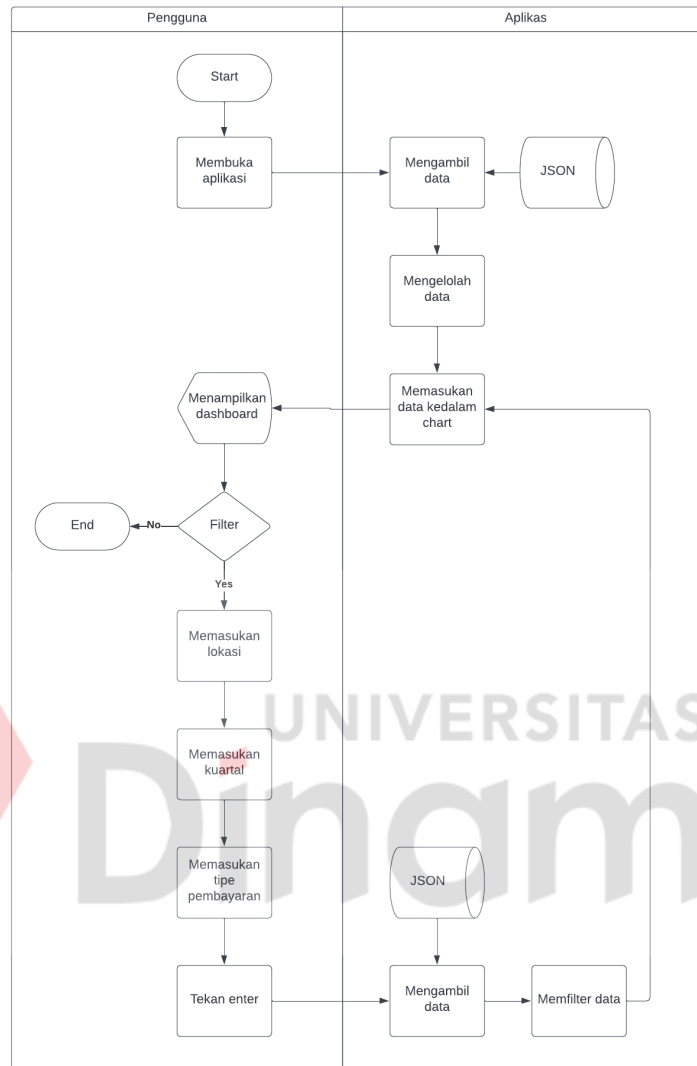
##### A. Fungsi *Dashboard* (Fgs-01)



Gambar 4. 1 Alur sistem fungsi *dashboard*

Pada gambar diatas dijelaskan bahwa alur dimulai dengan pengguna membuka aplikasi, lalu sistem akan mengambil *data* dari JSON, setelah mengambil *data* akan diolah, proses pengolahan sudah selesai dilanjutkan dengan memasukkan *data* olahan tersebut ke dalam *chart* yang akan ditampilkan kepada pengguna.

## B. Fungsi *filter* (Fgs-01)

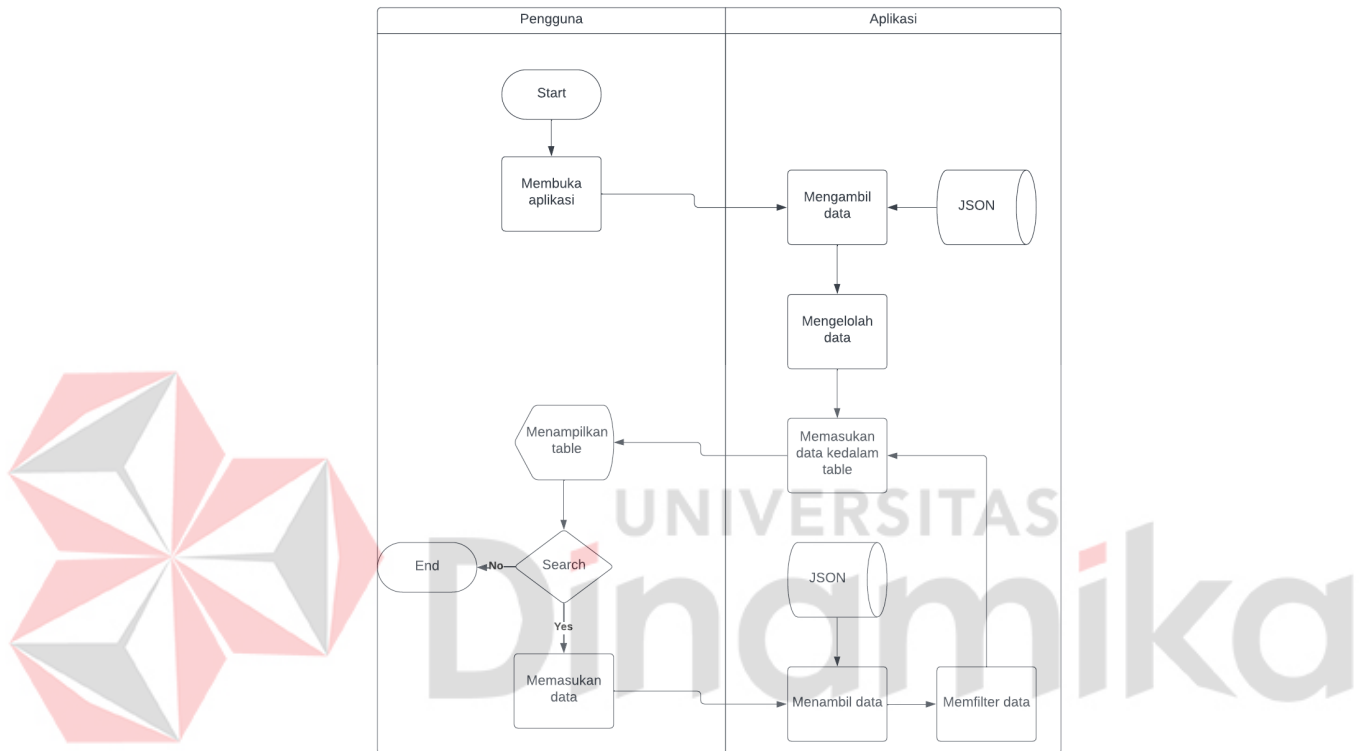


Gambar 4. 2 Alur sistem fungsi *filter*

Dapat dilihat pada gambar diatas alur sistem *filter* kurang lebih sama dengan alur *dashboard* namun ditambahi dengan percabangan apakah mau melakukan *filter* jika iya maka pengguna akan memasukan lokasi, tipe pembayaran, dan kuartal tahun dan menekan tombol *submit* Setelah menekan tombol *submit* maka sistem akan mengambil *data* lagi dari json, dan akan melakukan proses *filter*

sesuai dengan kriteria yang dimasukkan, setelah disaring *data* akan dimasukkan ke dalam *chart* dan ditampilkan kepada *user*.

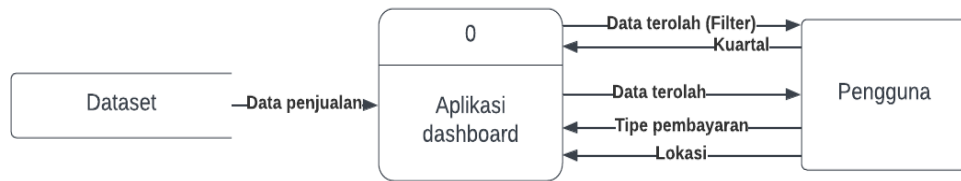
### C. Fungsi tabel (Fgs-03)



Gambar 4. 3 Alur sistem fungsi tabel

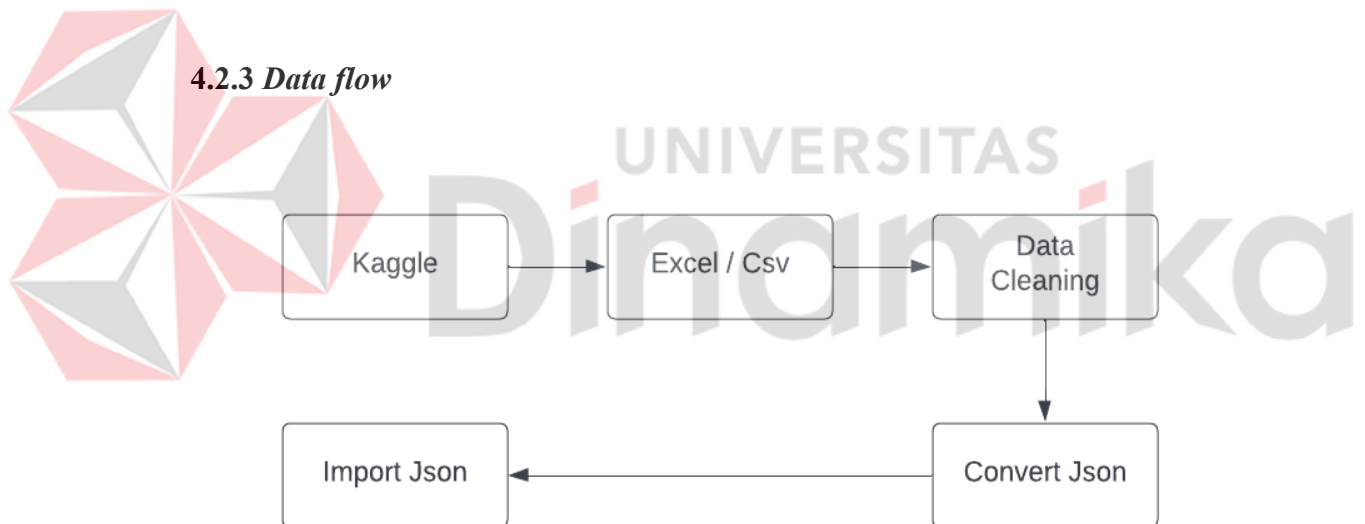
Proses alur sistem tabel *data* juga sama dijelaskan pada gambar diatas, dengan *filter* namun pengguna hanya memasukkan satu *data* saja untuk melakukan proses pencarian berbeda dengan *filter* yang memiliki 3 kategori.

### 4.2.2 Context diagram



Gambar 4. 4 Context diagram

*Context diagram* menjelaskan bagaimana alur *data* pada aplikasi, mulai dari *data* pengguna, dan *dataset* yang digunakan. Berikut diagram yang menjelaskan hal tersebut. Alur *data* tersebut dapat dilihat pada gambar diatas.



Gambar 4. 5 Data flow

Berikut penjelasan masing-masing tahap gambar diatas :

1. Kaggle

*Data* didapatkan berasal dari *website* Kaggle. Kaggle adalah *platform* yang menyediakan *dataset* dan kompetisi pemodelan prediktif.

2. Excel / Csv

Setelah mendownload *data* dari Kaggle maka *data* akan diubah menjadi excel / csv. Proses convert tersebut dilakukan di *website* yang menyediakan layanan *convert tipe file*.

### 3. *Data cleaning*

Setelah *file* diubah menjadi excel / csv maka *data* akan dibersihkan melalui aplikasi office excel. *Data* yang harus dibersihkan adalah sebagai berikut :

#### a. *Data duplikat*

*Data Duplikasi*: Identifikasi dan hapus baris *data* yang memiliki nilai yang sama pada semua kolom. Duplikasi dapat memengaruhi analisis dan menghasilkan hasil yang salah.

#### b. *Data pencilan*

*Outliers* (Pencilan): Cek apakah ada nilai yang sangat berbeda dari nilai lainnya. Pencilan dapat memengaruhi statistik dan model yang dibangun.

#### c. *Data kosong*

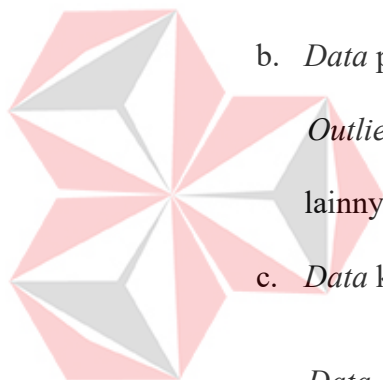
*Data Missing* (Hilang): Periksa apakah ada nilai yang hilang (*null*) pada kolom-kolom tertentu. Anda dapat mengisi nilai yang hilang atau menghapus baris yang memiliki nilai *null*.

### 4. *Convert JSON*

Setelah *data* dibersihkan langkah selanjutnya adalah mengubah *data* excel / csv tersebut menjadi json, agar bisa diolah oleh aplikasi.

### 5. *Import JSON*

Jika *data* Json sudah di konversi maka selanjutnya dapat diolah dengan menggunakan *javascript* yang akan dijadikan visualisasi dan *data* tabel.



UNIVERSITAS  
Dinamika

#### 4.2.4 Pemodelan *data*

*Data* yang digunakan dalam pembuatan aplikasi ini adalah tipe *data* JSON. Berbeda dengan sebuah *database* json disimpan dalam sebuah *file* yang memiliki karakteristik selayaknya *objek* dan *array* namun perbedaannya json adalah sebuah *file* penyimpanan *data* yang bisa digunakan oleh bermacam-macam bahasa. Tidak seperti *objek* dan *array* yang setiap bahasanya memiliki perintah / *code* tersendiri.

Berikut gambar untuk struktur *data* Json yang digunakan beserta properti dan *valuenya*.



```

[[
  "Status": "Unlinked",
  "Device_ID": "VJ300320609",
  "Location": "GuttenPlans",
  "Machine": "GuttenPlans_x1367",
  "Product": null,
  "Category": null,
  "Transaction": "15469028445",
  "TransDate": "2022-08-02",
  "Type": "Cash",
  "RCoil": "133",
  "RPrice": "1.5",
  "ROty": "1",
  "MCoil": "133",
  "MPrice": null,
  "MOty": "1",
  "LineTotal": "1.5",
  "TransTotal": "1.5",
  "Prcd_Date": "2022-08-02"
]]

```

Gambar 4. 6 Struktur *data* JSON

Pada *data* Json diatas kita bisa melihat ada 18 properti yang ada dalam *objek* json tersebut, masing masing properti akan menyimpan sebuah nilai yang memiliki tipe *data string*, dan ada properti yang tidak memiliki *data* / *null*. Berikut penjelasan setiap properti pada json diatas:

- *Status*: Menunjukkan apakah *data* mesin berhasil diproses atau tidak.



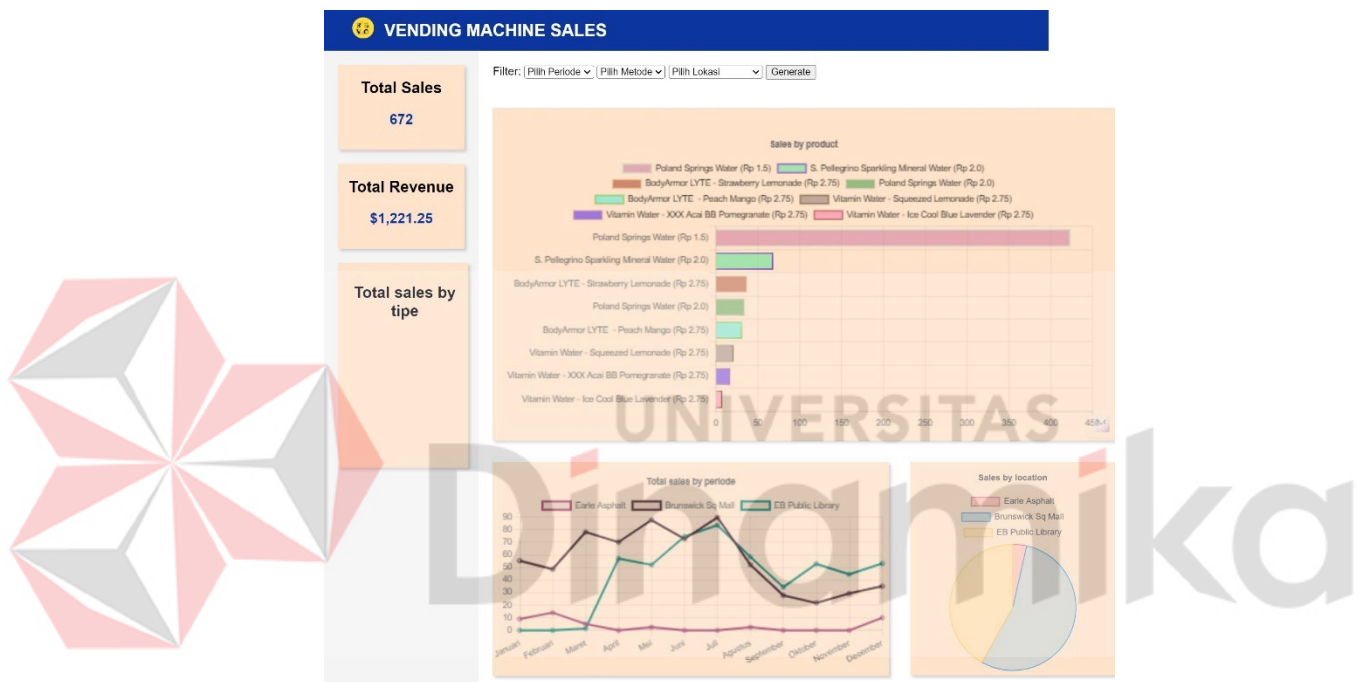
- *Device ID*: Pengidentifikasi elektronik unik (juga disebut ePort) untuk mesin penjual. Setiap mesin diberikan ePort \* device yang unik.
- *Location*: Menunjukkan lokasi mesin penjual.
- *Machine*: Nama mesin yang ramah pengguna.
- *Product*: Produk yang dijual dari mesin.
- *Category*: Kategori produk (Karbonasi / Makanan / Non-karbonasi / Air).
- *Transaction*: Pengidentifikasi unik untuk setiap transaksi.
- *TransDate*: Tanggal dan waktu transaksi.
- *Type*: Jenis transaksi (Tunai / Kredit).
- *RCoil*: Nomor koil yang digunakan untuk menjual produk.
- *RPrice*: Harga produk.
- *RQty*: Jumlah produk yang terjual. Biasanya satu, tetapi mesin dapat dikonfigurasi untuk menjual lebih banyak barang dalam satu transaksi.
- *MCoil*: Nomor koil yang dipetakan yang digunakan untuk menjual produk (dari toucan).
- *MPrice*: Harga produk yang dipetakan.
- *MQty*: Jumlah produk yang dipetakan terjual. Biasanya satu, tetapi mesin dapat dikonfigurasi untuk menjual lebih banyak barang dalam satu transaksi.
- *LineTotal*: Total penjualan per transaksi.
- *TransTotal*: Mewakili total semua transaksi yang akan muncul di Kartu Kredit. Seorang pengguna bisa membeli minuman seharga \$3 dan makanan ringan seharga \$1.5 sehingga totalnya menjadi \$4.50.
- *Prcd Date*: Tanggal ketika transaksi diproses oleh SeedLive (sebuah entitas yang digunakan untuk mengumpulkan semua transaksi secara elektronik).

### 4.3 Implementation

Berikut penerapan fungsi-fungsi yang dibutuhkan oleh pengguna dalam bentuk tampilan *website*.

#### 4.3.1 Fungsi dashboard (Fgs-01)

Tampilan dari fungsi Fgs-01 dapat dilihat pada gambar dibawah ini.



Gambar 4.7 Dashboard

Tampilan setiap visualisasi / *chart* (sub fungsi) dapat dilihat pada gambar dibawah ini :

- a. Sub fungsi visualisasi total penjualan per produk dan harga (Fgs-dsb-01)



Gambar 4. 8 Sub fungsi total penjualan per produk dan harga

Fungsi total penjualan untuk setiap produk bisa dilihat pada gambar diatas.

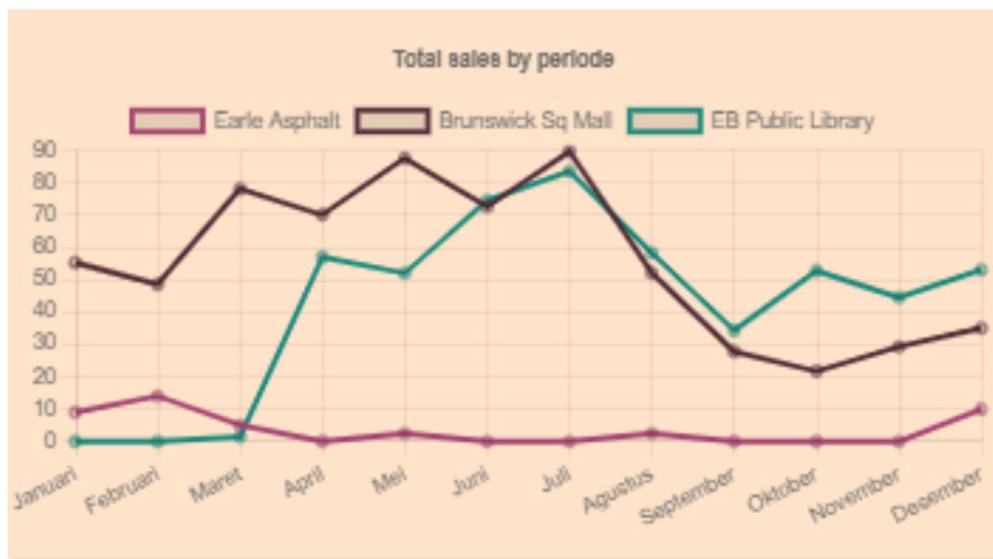
b. Sub fungsi visualiasi total transaksi per tipe pembayaran (Fgs-dsb-02)



Gambar 4. 9 Sub fungsi total penjualan setiap tipe pembayaran

Pada gambar bisa kita lihat bahwa memvisualisasikan total penjualan berdasarkan tipe pembayaran.

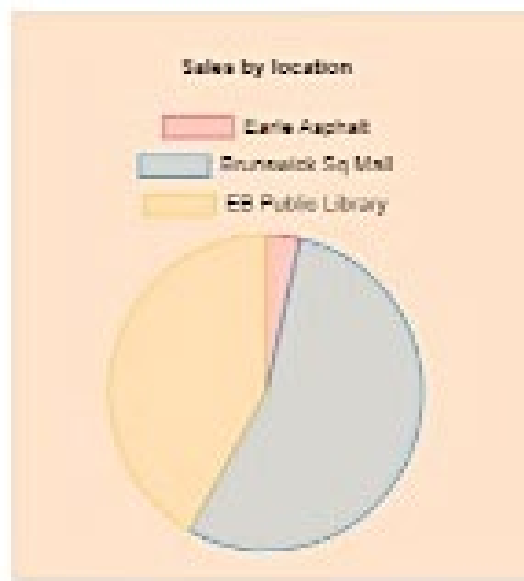
- c. Sub fungsi Visualisasi total penjualan perbulan untuk setiap toko (Fgs-dsb-03)



Gambar 4. 10 Sub fungsi total penjualan perbulan

*Chart* fungsi yang menuntukan total penualan perbulan dapat dilihat pada gambar diatas.

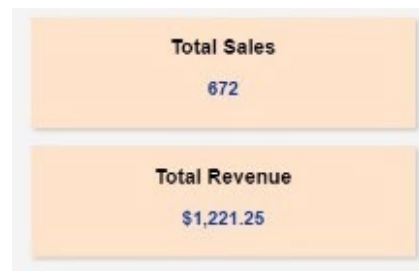
- d. Sub fungsi Visualiasi total penjualan perlokasi (Fgs-dsb-04)



Gambar 4. 11 Sub fungsi total penjualan setiap lokasi

Visualisasi fungsi total penjualan dapat kita lihat pada gambar diatas.

- e. Sub fungsi total penjualan dan pendapatan keseluruhan  
(Fgs-dsb-05 dan Fgs-dsb-06)



Gambar 4. 12 Total penjualan seluruh pendapatan dan penjualan

UI tampilan kedua fungsi pada fungsi total penjualan dan pendapatan (Fgs-dsb-05 dan Fgs-dsb-06) dapat dilihat pada gambar diatas.

#### 4.3.2 Fungsi filter (Fgs-02)

Untuk tampilan fungsi *filter* (Fgs-02) dapat dilihat pada gambar dibawah.

Filter:

Gambar 4.13 *Filter*

#### 4.3.3 Fungsi tabel (Fgs-03)

Tampilan fungsi tabel (Fgs-03) bisa dilihat pada gambar di bawah ini.

Status	Device ID	Location	Machine	Product	Category	Transaction	TransDate	Type	RPrice	RQty	LineTotal	TransTo
Processed	VJ300320686	Earle Asphalt	Earle Asphalt x1371	BodyArmor LYTE - Berry Punch	Water	16141024168	2022-12-21	Cash	2.5	1	2.5	2.5
Processed	VJ300320686	Earle Asphalt	Earle Asphalt x1371	BodyArmor LYTE - Berry Punch	Water	16143744067	2022-12-21	Credit	2.5	1	2.5	2.5
Processed	VJ300320686	Earle Asphalt	Earle Asphalt	BodyArmor LYTE -	Water	14542825549	2022-01-08	Cash	2.0	1	2.0	2.0

Showing 1 to 10 of 657 entries

Previous 1 2 3 4 5 ... 66 Next

Gambar 4.14 Tabel

#### 4.4 Testing

Proses pengujian fungsionalitas aplikasi akan menggunakan metode *Blackbox*, yang menilai fungsi aplikasi tanpa memeriksa struktur *internalnya*. Metode ini memungkinkan penguji untuk memastikan setiap fitur berfungsi sesuai spesifikasi.

Tabel pengujian akan mencakup kolom-kolom berikut:

1. **Test Scenario ID**: Identifikasi unik untuk setiap skenario pengujian.
2. **Test Scenario Description**: Deskripsi rinci skenario pengujian.
3. **Test Case ID**: Identifikasi unik untuk setiap kasus pengujian.
4. **Test Case Description**: Deskripsi rinci setiap scenario.
5. **Step**: Langkah-langkah pengujian terperinci.
6. **Expected Result**: Hasil yang diharapkan berdasarkan spesifikasi.
7. **Actual Result**: Hasil aktual dari pengujian.
8. **Status**: Status akhir pengujian, seperti "Lulus" atau "Gagal."

Pengujian dianggap berhasil jika hasil aktual sesuai dengan yang diharapkan. Metode *Blackbox* ini fokus pada pengalaman pengguna akhir dan memastikan aplikasi memenuhi kebutuhan tanpa mempengaruhi implementasi *internal*.

Berikut hasil *testing* aplikasi dalam bentuk tabel:

Tabel 4. 3 Tabel hasil *test* aplikasi

<i>Test Scenario ID</i>	<i>Test Scenario Description</i>	<i>Test Case Id</i>	<i>Test Case Description</i>	<i>Step</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
TS_DB_01	Verify filter dashboard	TC_DB_Filter_001	Tidak memilih lokasi, kuartal dan tipe pembayaran	1. Biarkan semua <i>filter</i> tidak dalam keadaan terpilih 2. Tekan <i>submit filter</i>	Muncul anda belum memilih <i>filter</i>	Muncul anda belum memilih <i>filter</i>	Pass
		TC_DB_Filter_002	Tidak memilih lokasi, memilih kuartal dan memilih tipe pembayaran	1. Biarkan <i>filter</i> lokasi dalam keadaan tidak dipilih. 2. Pilih kuartal. 3. Pilih pembayaran. 3. Tekan <i>submit filter</i>	Muncul anda belum memilih <i>filter</i>	Muncul anda belum memilih <i>filter</i>	Pass

<i>Test Scenario ID</i>	<i>Test Scenario Description</i>	<i>Test Case Id</i>	<i>Test Case Description</i>	<i>Step</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
		TC_DB_Filter_003	Memilih lokasi, tidak memilih kuartal dan tidak memilih tipe pembayaran	1. Pilih lokasi. 2. Biarkan <i>filter</i> kuartal dan tipe pembayaran dalam keadaan tidak dipilih. 3. Tekan <i>submit</i>	Muncul anda belum memilih <i>filter</i>	Muncul anda belum memilih <i>filter</i>	Pass
		TC_DB_Filter_004	Memilih lokasi, memilih kuartal dan tidak memilih tipe pembayaran	1. Pilih lokasi. 2. Pilih kuartal. 3. Biarkan <i>filter</i> tipe pembayaran dalam keadaan tidak dipilih. 4. Tekan <i>submit</i>	Muncul anda belum memilih <i>filter</i>	Muncul anda belum memilih <i>filter</i>	Pass
		TC_DB_Filter_005	Memilih lokasi, tidak memilih kuartal dan memilih tipe pembayaran	1. Pilih lokasi. 2. Biarkan <i>filter</i> kuartal dalam keadaan tidak dipilih. 3. Pilih tipe pembayaran. 3. Tekan <i>submit</i>	Muncul anda belum memilih <i>filter</i>	Muncul anda belum memilih <i>filter</i>	Pass



<i>Test Scenario ID</i>	<i>Test Scenario Description</i>	<i>Test Case Id</i>	<i>Test Case Description</i>	<i>Step</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
		TC_DB_Filter_006	Memilih lokasi, memilih kuartal dan memilih tipe pembayaran	1. Pilih lokasi. 2. Pilih kuartal. 3. Pilih pembayaran. 4. Tekan <i>submit</i>	Menampilkan <i>chart</i> sesuai dengan filternya	Menampilkan <i>chart</i> sesuai dengan filternya	Pass
TS_DB_002	<i>Verify search tabel functionality</i>	TC_DB_tabelSearch_001	Mencari <i>data</i> yang tidak ada ( <i>invalid data</i> )	1. Masukkan data yang tidak ada	Tabel tidak akan menampilkan data karena data tidak ada	Tabel tidak akan menampilkan data karena data tidak ada	Pass
		TC_DB_tabelSearch_002	Mencari <i>data</i> ada ( <i>valid data</i> )	1. Masukkan data yang ada	Tabel menampilkan <i>data</i>	Tabel menampilkan <i>data</i>	Pass
TS_DB_004	<i>Verify sorting tabel functionality</i>	TC_DB_tabelSorting_001	Mengaktifkan fitur <i>sorting</i>	1. Pilih kolom tabel yang akan di <i>sorting</i> 2. Tekan tombol panah yang ada dikolom	Tabel akan menampilkan <i>data</i> dari yang terbesar sampai yang terkecil	Tabel akan menampilkan <i>data</i> dari yang terbesar sampai yang terkecil	Pass

<i>Test Scenario ID</i>	<i>Test Scenario Description</i>	<i>Test Case Id</i>	<i>Test Case Description</i>	<i>Step</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Status</i>
		TC_DB_tabelSorting_002	Mematikan fitur <i>sorting</i>	1. Tekan tombol panah yang sudah diaktifkan.	Tabel akan kembali seperti semula	Tabel akan kembali seperti semula	Pass



UNIVERSITAS  
**Dinamika**

#### 4.5 Maintenance

Pemeliharaan aplikasi merupakan fase penting setelah aplikasi diluncurkan. Tujuannya adalah menjaga kinerja optimal aplikasi dalam jangka panjang. Beberapa aktivitas utama dalam pemeliharaan aplikasi meliputi:

1. **Perbaikan *error*:** Menangani masalah yang muncul untuk memastikan pengalaman pengguna yang lancar.
2. **Pembaruan sistem:** Mengimplementasikan peningkatan fitur dan perbaikan keamanan secara berkala.
3. **Layanan pemeliharaan pengguna:** Memberikan dukungan teknis kepada pengguna yang mengalami masalah.
4. **Pemantauan kinerja:** Memantau kecepatan, stabilitas, dan penggunaan sumber daya untuk mengidentifikasi potensi masalah.
5. **Penyesuaian keamanan:** Meningkatkan keamanan aplikasi secara berkala untuk melindungi *data* pengguna.

Dengan melakukan pemeliharaan rutin, pengembang dapat memperpanjang umur aplikasi, mengurangi risiko kegagalan, dan meningkatkan kepuasan pengguna. Pemeliharaan aplikasi yang efektif juga berperan dalam menjaga reputasi pengembang dan membangun kepercayaan pengguna, yang pada akhirnya akan mendorong pertumbuhan pengguna baru.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Kesimpulan yang didapatkan dari pengembangan aplikasi *dashboard* berbasis web ini ialah sebagai berikut :

1. Aplikasi berjalan dengan lancar semua fitur / kebutuhan yang dibutuhkan oleh pengguna dapat berjalan dengan baik.
2. Aplikasi dapat segera digunakan untuk kebutuhan *monitoring* karena sudah dilakukannya deployment sehingga dapat segera digunakan.

#### 5.2 Saran

Berikut beberapa saran pengembangan yang bisa dilakukan kepada aplikasi kedepannya :

1. *Website* dapat ditambahkan fungsi CUD (*Create, Update, Delete*), sebuah *data*.  
Bisa ditambahkan semacam *form* untuk melakukan *input data, update* dan *delete* untuk menghapus jika ada *data* yang salah masuk dalam sistem
2. *Website* juga bisa ditambahkan fungsi *real time monitoring* dimana *website* akan dapat mendeteksi setiap detiknya bila ada transaksi dan *data* terbaru dapat ditampilkan dalam *website*.
3. Penyesuaian grafik juga diperlukan. Penyesuaian tersebut bisa penambahan grafik, pengurangan grafik, bila terdapat grafik yang kurang sesuai dan ada grafik yang dibutuhkan namun tidak ada.

## DAFTAR PUSTAKA

- Bach, B., Freeman, E., Abdul-Rahman, A., Turkay, C., Khan, S., Fan, Y., & Chen, M. (2023). Dashboard Design Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 29(1). <https://doi.org/10.1109/TVCG.2022.3209448>
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. In *International Journal of Engineering & Technology (iJET)* (Vol. 2, Issue 5). [http://iet-journals.org/archive/2012/may\\_vol\\_2\\_no\\_5/255895133318216.pdf](http://iet-journals.org/archive/2012/may_vol_2_no_5/255895133318216.pdf)
- Batalas, N., Khan, V. J., & Markopoulos, P. (2021). Executabel HTML. *SoftwareX*, 14. <https://doi.org/10.1016/j.softx.2021.100691>
- Bourhis, P., Reutter, J. L., & Vrgoč, D. (2020). JSON: Data model and query languages. *Information Systems*, 89. <https://doi.org/10.1016/j.is.2019.101478>
- Cao, Y., Ikenoya, Y., Kawaguchi, T., Hashimoto, S., & Morino, T. (2023). A Real-Time Application for the Analysis of Multi-Purpose Vending Machines with Machine Learning. *Sensors*, 23(4). <https://doi.org/10.3390/s23041935>
- Fitria Nur Hasanah, M. P., Rahmania Sri Untari, M. P., & Pd, M. (2020). *BUKU AJAR Rekayasa Perangkat Lunak* (S. Pd. , M. K. Mohammad Suryawinata, Ed.; 1st ed., Vol. 1). UMSIDA Press.
- Hasan, L., & Abuelrub, E. (2011). Assessing the quality of web sites. *Applied Computing and Informatics*, 9(1). <https://doi.org/10.1016/j.aci.2009.03.001>
- Ika Ratna Indra Astutik, S. Kom. , M. T., & Mochamad Alfian Rosid, S. K. M. Kom. (2019). *BUKU AJAR Pemrograman Berbasis Web* (M. P. Septi Budi Sartika & S. H. , M. Kn. M. Tanzil Multazam, Eds.; 1st ed., Vol. 1). UMSIDA Press.
- Syarif, M., & Pratama, E. B. (2021). Analisis Metode Pengujian Perangkat Lunak Blackbox Testing Dan Pemodelan Diagram Uml Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model Waterfall. *Jurnal Teknik Informatika Kaputama (JTİK)*, 5(2).
- Wirfs-Brock, A., & Eich, B. (2020). JavaScript: The first 20 years. *Proceedings of the ACM on Programming Languages*, 4(HOPL). <https://doi.org/10.1145/3386327>
- Zaidir, Z., Nuswantoro, B. S., Listiawan, I., Sahal, A., Diqi, M., & Meliala, D. A. (2023). Pengujian Software Pengendalian Penduduk Permanen-Nonpermanen Dengan BlackBox Test dan Evaluasi Penerimaan Metode Technology Acceptance Model. *Jurnal Teknologi Informasi Dan Terapan*, 10(1). <https://doi.org/10.25047/jtit.v10i1.305>