



## BAB II TINJAUAN PUSTAKA

### 2.1 Teori Umum

#### 2.1.1 Pengertian Komputer

Menurut [1] Komputer merupakan alat untuk mengolah data sesuai perintah yang sudah dirumuskan secara cepat dan tepat, serta diorganisasikan supaya secara otomatis menerima dan menyimpan data berdasarkan intruksi intruksi yang telah tersimpan didalam memori. Pada awalnya, kata komputer berarti orang berprofesi dalam bidang aritmatika, yaitu yang melakukan perhitungan aritmatika dengan menggunakan alat ataupun tidak. Tapi, arti kata ini mengalami perubahan. Istilah komputer mempunyai arti yang luas dan berbeda bagi setiap orang. Istilah komputer (*computer*) diambil dari bahasa latin *computere* yang berarti menghitung (*to compute atau to reckon*).

Berdasarkan definisi diatas, penulis menyimpulkan bahwa komputer mengandung arti hitung yang merupakan alat elektronik yang dapat melakukan pekerjaan manusia, seperti mengolah dan memproses data sesuai dengan rumus yang diperintahkan dan dapat menyimpan program dan hasil pengolahan serta bekerja secara otomatis.

#### 2.1.2 Pengertian Perangkat Lunak

Menurut [1] Perangkat Lunak adalah sekumpulan data elektronik yang berupa program. Program adalah sekumpulan intruksi yang digunakan untuk mengendalikan perangkat keras komputer. Perangkat lunak merupakan program program komputer yang berguna menjalankan suatu pekerjaan sesuai dengan yang dikehendaki.

Berdasarkan definisi di atas penulis menyimpulkan bahwa perangkat lunak adalah program komputer., prosedur yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara pengguna yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai.



---

### **2.1.3 Pengertian Data**

Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata data adalah keterangan yang benar dan nyata. Berdasarkan definisi diatas penulis menyimpulkan bahwa, pengertian data adalah kumpulan catatan dari fakta dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf-huruf yang dapat diolah dalam proses penelitian dan menghasilkan output yang menjawab masalah dan pernyataan penelitian.

### **2.1.4 Pengertian Basis Data**

Menurut [2] basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau *informasi*. Atau basis data (*database*) merupakan kumpulan dari data yang saling berhubungan (relasi) antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu.

Berdasarkan definisi di atas penulis menyimpulkan bahwa, basis data (*database*) adalah sekumpulan *informasi* yang terkomputerisasi yang disimpan dalam komputer secara sistematis dan diperiksa menggunakan suatu program komputer untuk memperoleh *informasi* dan memiliki tujuan utama memelihara data yang sudah diolah dan membuat *informasi* tersedia saat dibutuhkan.

## **2.2 Teori Judul**

### **2.2.1 Pengertian Aplikasi**

Menurut Kamus Besar Bahasa, mendefinisikan "aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Sedangkan menurut [3] aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikais mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang



---

biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Berdasarkan beberapa definisi diatas penulis dapat menyimpulkan bahwa, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

### **2.2.2 Pengertian Implementasi**

Menurut kamus besar Bahasa Indonesia, implementasi berarti pelaksanaan atau penerapan. Istilah implementasi sering dikaitkan dengan suatu kegiatan yang dilaksanakan untuk mencapai suatu tujuan tertentu. Implementasi juga didefinisikan sebagai salah satu upaya untuk mewujudkan suatu sistem. Tanpa implementasi, sebuah konsep tidak akan terwujud. Untuk mengimplementasikan suatu hal harus disertai sarana atau penunjang yang mendukung, karena nantinya hal tersebut akan menimbulkan dampak atau akibat terhadap sesuatu.

### **2.2.3 Pengertian Metode *Agile Kanban***

Metode *Agile Kanban* adalah manajemen proses visual yang dapat mengelola pengetahuan dan pekerjaan dengan mempertimbangkan pendekatan pengiriman *Just In Time* (JIT). JIT tidak membebani anggota tim karena metode *Kanban* berfokus pada menghilangkan kemacetan dan pemborosan, serta mengurangi waktu tunggu, yang mengarah pada peningkatan dari jumlah keluaran [4].

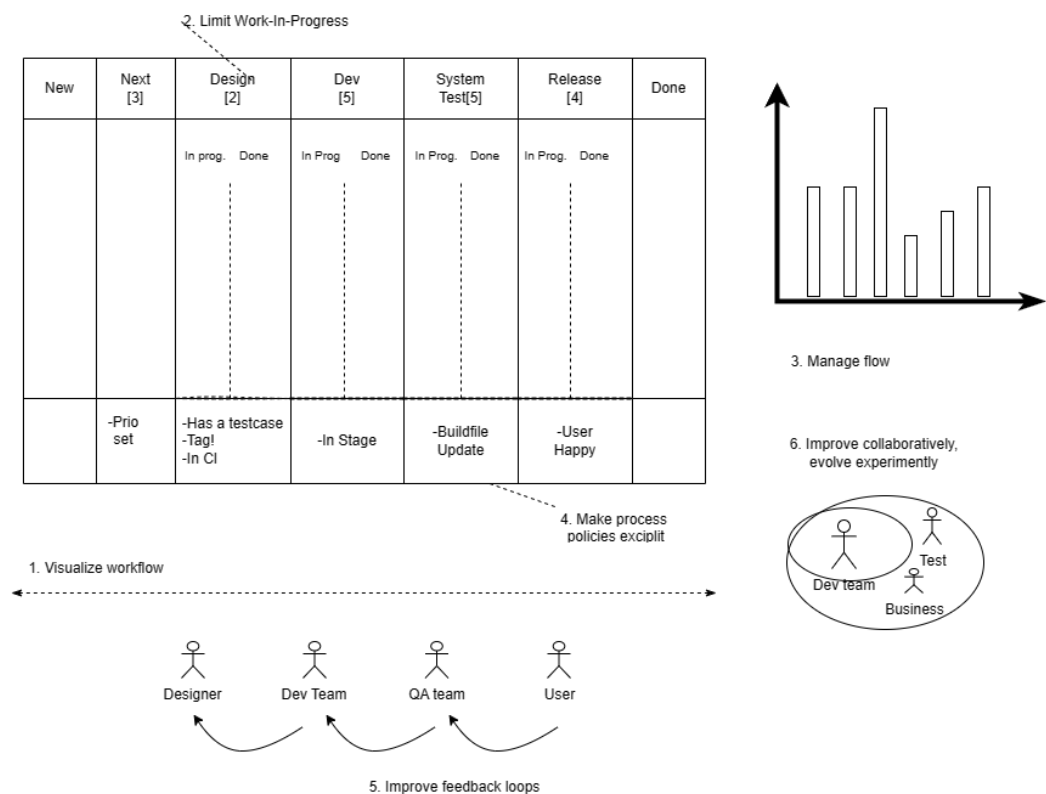
Metode *Kanban* merupakan yang digunakan dalam proses manufaktur maupun proses pengembangan perangkat lunak untuk meningkatkan efisiensi dan produktivitas proyek agar alur manajemen proyek terarah dan sesuai dengan waktu serta anggaran yang telah diinginkan [5].

Berdasarkan definisi di atas, penulis menyimpulkan bahwa Metode *Agile Kanban* merupakan sebuah metode pengembangan perangkat lunak yang menggunakan pendekatan *Just in Time* (JIT) sehingga meningkatkan efisiensi dan produktivitas pengembangan proyek.



*Kanban* pada awalnya adalah alat yang digunakan oleh Toyota untuk menyeimbangkan permintaan dan kapasitas di seluruh rantai nilai. Idennya sederhana: kartu *Kanban* dikirim ke hulu ketika ada kebutuhan untuk suku cadang. Hanya dengan begitu produksi untuk apa yang diperlukan dilakukan oleh sejumlah bagian. Kedatangan kartu baru adalah sinyal untuk menghasilkan lebih banyak bagian, dan kurangnya kartu adalah sinyal untuk berhenti. Jumlah kartu terbatas untuk mencegah kelebihan produksi dan mengurangi bagianbagian yang diperlukan dalam produksi. Metode ini mempertahankan bagian setengah jadi mengikat modal kerja yang berharga yang dapat digunakan sebaliknya untuk investasi [6].

Berikut adalah mengenai the six core practices of *Kanban* apply to knowledge yang dijelaskan sebagai berikut:



**Gambar 2.1** The six core practices of *Kanban* apply to knowledge work

(Sumber: Skarin, 2015:6)

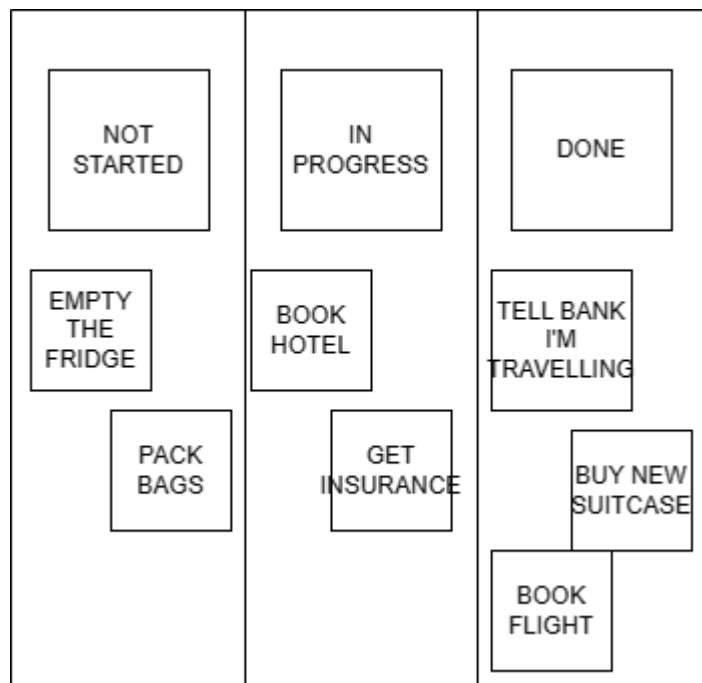
Yang paling sederhana dan beberapa pendapat tentang *Kanban board*, terdiri dari tiga kolom: *things to do*, *tasks in progress* and *finally work done*. Form



at sederhana ini sangat cocok untuk setiap proyek atau alur kerja. Tahap tahap yang disarankan adalah *ideas, to do, doing and done*. Diantara setiap status *Kanban* jelas karena ada perpindahan kartu *Kanban* dari kolom *ideas, to do, doing and done*: [6]

1. Ideas - suatu pemikiran atau gagasan yang dipikirkan melalui pemikiran yang mungkin atau tidak mungkin terjadi pada suatu pertanyaan.
2. To do or in progress - ketika sebuah ide dipikirkan dan satu-satunya pertanyaan adalah siapa yang akan melakukan pekerjaan dan kapan akan dimulai.
3. In progress - setelah seseorang melakukan suatu tugas tersebut dan masih dikerjakan.
4. Done - hasil pekerjaan yang telah selesai.

Berikut ini adalah *format sederhana Kanban Board* dalam berbagai proyek alur kerja:



**Gambar 2.2** Format sederhana *Kanban Board*  
(Sumber: Cole, 2015:74)



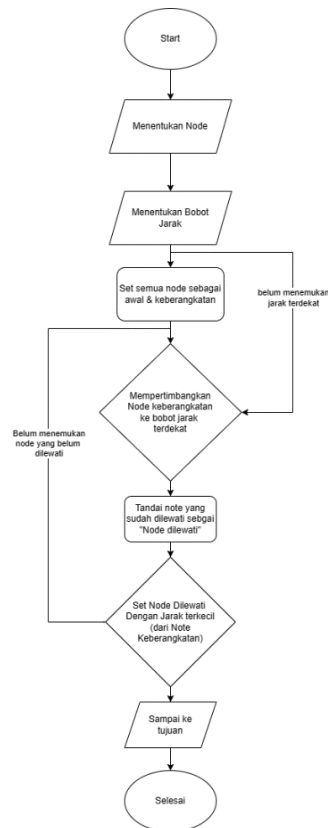
#### 2.2.4 Pengertian Rute Terpendek

Rute terpendek (*shortest path*) adalah Masalah dari mencari jarak terpendek antara dua persimpangan dari peta jalan (simpul graf yang berhubungan ke persimpangan dan ujung yang berhubungan ke segmen jalan, yang tiap-tiap nya diberi bobot oleh panjang dari segmen jalan) dapat dimodelkan dari kasus spesial dari masalah jarak terpendek dalam graf.

#### 2.2.5 Pengertian Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah salah satu algoritma dalam mencari rute. Pada algoritma *Dijkstra* juga termasuk dalam algoritma untuk masalah pencarian graf yang mampu menuntaskan masalah mencari lintasan terpendek dengan satu sumber pada sebuah graf yang digunakan untuk menyelesaikan masalah lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki cost sisi negatif, dan menghasilkan sebuah alur lintasan terpendek. Algoritma *Dijkstra* ini sering digunakan untuk melakukan routing [7].

Algoritma *Dijkstra* ini sendiri ditujukan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari titik keberangkatan ke titik lainnya. Misalkan sebuah gedung dan monumen dijadikan titik dan jalanan dijadikan garis, maka algoritma *dijkstra* akan melakukan perhitungan terhadap semua garis dengan bobot terkecil dari setiap titik [7].



**Gambar 2.3** Algoritma *Dijkstra*  
(sumber: Aldy Cantona, 2020)

Dalam tahapan algoritma *Dijkstra* ini dapat dilakukan dengan tahapan berikut:

1. Tentukan titik awal dimana titik tersebut akan menjadi *node* pertama atau *node* awal, lalu tentukan bobot jarak pada *node* awal ke *node* terdekat satu demi satu, *Dijkstra* akan melakukan *developing* untuk mencari bobot terkecil dari satu *node* ke *node* lainnya dan ke *node* selanjutnya satu per satu.
2. Tentukan bobot (jarak) dari setiap *node* ke *node* lainnya, kemudian set nilai 0 pada *node* pertama dan nilai tak terhingga pada *node* lainnya. Setting semua *node* yang belum dilalui dan setting *node* awal sebagai “*node* keberangkatan”
3. Pada *node* keberangkatan, perhitungkan *node* lainnya yang paling dekat dengan keberangkatan yang belum dilalui dan perhitungkan jarak dari awal



keberangkatan. Jika jaraknya lebih kecil dari jarak sebelumnya (yang telah diperhitungkan sebelum-nya) hapus data yang lama simpan ulang data jarak sebelumnya dengan data jarak yang lebih pendek.

4. Setelah kita selesai mengitung dan memper-timbangan setiap jarak pada *node* lainnya yang berdekatan, tandai *node* yang sudah dilalui sebagai “*Node dilewati*”. *Node* yang sudah dilewati tidak akan pernah dicek kembali, jarak yang akan disimpan adalah jarak dengan bobot paling kecil bobotnya,
5. Setting “*Node belum dilewati*”, dengan bobot jarak paling kecil (dari *node* keberangkatan) sebagai “*Node Keberangkatan*” untuk keberangkatan selanjutnya dan ulangi langkah.

### 2.2.6 Pengertian GIS

GIS (*Geographic Information System*) atau Sistem Informasi Geografis (SIG) yakni sistem yang dibuat oleh komputer yang memungkinkan suatu algoritma berfungsi untuk mengidentifikasi suatu area di permukaan bumi. Lebih lanjut, Sistem Informasi Geografis (SIG) menyediakan data spasial dan non-spasial kejadian permukaan bumi secara alami, data vektor yang memberikan informasi keberadaan suatu posisi [8].

Data spasial merupakan data yang berhubungan dengan suatu objek di permukaan bumi. *Database* spasial merupakan penyimpanan data yang terstruktur dalam sebuah sistem basis data [8].

## 2.3 Teori Khusus

### 2.3.1 Unified Modelling Language (UML)

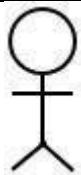
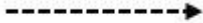


UML adalah sebuah patokan bahasa untuk menulis kerangka kerja terperinci dari sebuah perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun dan mendokumentasikan sebuah sistem perangkat lunak.[9]



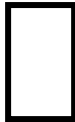


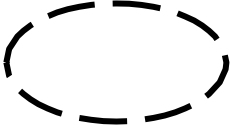
a. *Use case Diagram*

*Use case* diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.[10]

**Tabel 2.1** Simbol – simbol *Use case*

NO	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<b>Dependency</b>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek.
4	<<include>>	<b>Include</b>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5	<<extend>>	<b>Extend</b>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<b>Association</b>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Lanjutan Tabel 2.1 Simbol – simbol *Use case*

NO	Gambar	Nama	Keterangan
7		System	Menspesifikasikan paket yang menampilkan system secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
10		Collaboration	Interaksi aturan – aturandan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen – elemennya (sinergi).


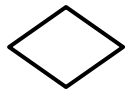
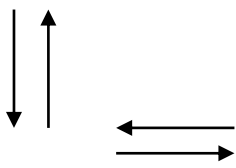
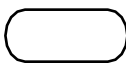


Sumber: Purwanto (2019)



b. *Activity diagram*

*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses – proses apa saja yang terjadi pada sistem.[10]

**Tabel 2.2** Simbol – simbol *Activity diagram*

NO	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing – masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
3		<b>Line Connector</b>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.
4		<b>Action</b>	State dari system yang mencerminkan eksekusi dari suatu aksi
5		<b>Initial Node</b>	Bagaimana objek dibentuk dan diawali.
6		<b>Activity Final Node</b>	Bagaimana objek dibentuk dan diakhiri

Sumber: Purwanto (2019)



### c. Sequence Diagram

*Sequence* diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case* diagram.[10]

**Tabel 2.3** Simbol – simbol *Sequence* Diagram

NO	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem.
4		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel
5		<i>A Focus Of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i> .
6		<i>A Message</i>	Menggambarkan pengiriman pesan.

Sumber : Purwanto (2019)

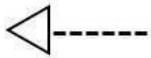
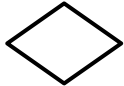
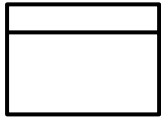
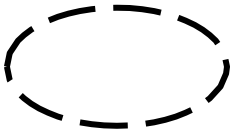
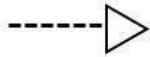

### d. Class diagram

*Class diagram* yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas – kelas maupun paket – paket yang ada pada suatu sistem yang



nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi – relasi yang terdapat pada sistem tersebut.[10]

**Tabel 2.4** Simbol – simbol *Class diagram*

NO	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek.
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek – objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	System yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5		<i>Realization</i>	Operasi yang benar – benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent).

Sumber : Purwanto (2019)







### 2.3.2 Flowchart

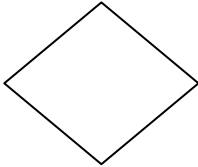
Menurut Indrajani dalam Rusmawan (2019:48), mengatakan bahwa “*Flowchart* atau bagan alir merupakan gambaran secara fisik dari langkah-langkah dan urutan prosedur suatu program”. Sedangkan menurut Sari dan Indra (2021:15), mengatakan bahwa “*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan proses urutan program secara mendetail”.

Berdasarkan beberapa definisi diatas penulis dapat menyimpulkan bahwa, *Flowchart* (Diagram Alur) adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan proses urutan program secara mendetail dengan representasi grafis dari sistem yang mendeskripsikan relasi fisik antar entitas-entitas intinya.

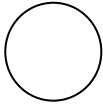
**Tabel 2.5** Tabel simbol-simbol Diagram Alir (*Flowchart*)

No	Simbol	Nama	Keterangan
1.		Terminator	Simbol ini digunakan untuk menyatakan titik awal atau titik akhir diagram alir.
2.		Proses	Simbol ini digunakan untuk menyatakan sebarang proses.
3.		Proses terdefinisi	Simbol ini menyatakan prosedur lain yang telah didiagramalirkan pada tempat lain.
4.		<i>Input/output</i> atau kadang disebut data	Simbol ini menyatakan operasi pemasukan data atau penampilan data.



5.		Keputusan	Simbol ini digunakan untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupa suatu pernyataan yang jawabannya berupa dua kemungkinan, yaitu “ya” atau “tidak”.
----	---	-----------	---

**Lanjutan Tabel 2.5** Tabel simbol-simbol Diagram Alir (*Flowchart*)

6.		Konektor	Simbol ini digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir.
----	---	----------	---

Sumber: Kadir 2017



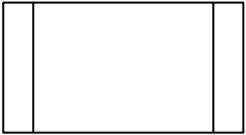
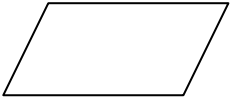
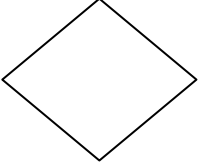
### 2.3.3 Kamus Data

Menurut Indrajani dalam Rusmawan (2019:48), mengatakan bahwa “*Flowchart* atau bagan alir merupakan merupakan gambaran secara fisik dari langkah-langkah dan urutan prosedur suatu program”. Sedangkan menurut Sari dan Indra (2021:15), mengatakan bahwa “*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan proses urutan program secara mendetail”.

Berdasarkan beberapa definisi diatas penulis dapat menyimpulkan bahwa, *Flowchart* (Diagram Alur) adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan proses urutan program secara mendetail dengan representasi grafis dari sistem yang mendeskripsikan relasi fisik antar entitas-entitas intinya.

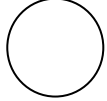


**Tabel 2.6** Tabel simbol-simbol Diagram Alir (*Flowchart*)

No	Simbol	Nama	Keterangan
1.		Terminator	Simbol ini digunakan untuk menyatakan titik awal atau titik akhir diagram alir.
2.		Proses	Simbol ini digunakan untuk menyatakan sebarang proses.
3.		Proses terdefinisi	Simbol ini menyatakan prosedur lain yang telah didiagramalirkan pada tempat lain.
4.		<i>Input/output</i> atau kadang disebut data	Simbol ini menyatakan operasi pemasukan data atau penampilan data.
5.		Keputusan	Simbol ini digunakan untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupa suatu pernyataan yang jawabannya berupa dua kemungkinan, yaitu “ya” atau “tidak”.



**Lanjutan Tabel 2.6** Tabel simbol-simbol Diagram Alir (*Flowchart*)

6.		Konektor	Simbol ini digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir.
----	---	----------	---

Sumber: Kadir 2017

## 2.4 Teori Program

### 2.4.1 Pengertian Hypertext Preprocessor (PHP)

Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat website yang bersifat *server-side scripting*. PHP bersifat dinamis. PHP dapat dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan Mac Os. Selain Apache, PHP juga mendukung beberapa web server lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan *database* untuk menghasilkan halaman web yang dinamis



**Gambar 2.4** Logo PHP  
(Sumber: <https://www.php.net/>)

Sistem manajemen *database* yang sering digunakan bersama PHP adalah MYSQL. Namun, PHP juga mendukung sistem manajemen *Database* Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL.



#### 2.4.2 Pengertian XAMPP



**Gambar 2.5** Logo XAMPP

(Sumber: <https://www.apachefriends.org/>)

XAMPP adalah perangkat lunak open source, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain: Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

#### 2.4.3 Pengertian Codeigniter



**Gambar 2.6** Logo Codeigniter

(Sumber: <https://codeigniter.com/>)

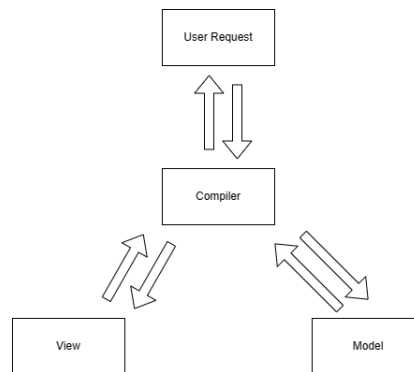
Menurut [11] *framework* codeigniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP, dengan paradigma *Model-view-controller* (MVC).

#### 2.4.4 Pengertian *Model-view-controller* (MVC)

Menurut [11], *model-view-controller* adalah *framework* PHP yang dibuat berdasarkan kaidah *model-view-controller*. Dengan MVC, maka memungkinkan



pemisahan antara layer application-logic dan presentation. Sehingga, dalam sebuah pengembangan web, seorang programmer bisa berkonsentrasi pada *core-system*. Adapun alur program aplikasi berbasis *framework* codeigniter dapat dilihat pada gambar di bawah ini.



**Gambar 2.7** Model MVC  
(Sumber: Yeni Anggraini, 2020)

Gambar 2.7 menerangkan bahwa ketika datang sebuah user request, maka akan ditangani oleh controller, kemudian controller akan memanggil *model* jika memang diperlukan operasi *database*. Hasil dari query oleh *model* akan dikembalikan ke controller. Setelah itu controller akan memanggil view yang tepat dan mengkombinasikannya dengan hasil query *model*. Hasil akhir dari operasi ini akan ditampilkan di browser. Dalam konteks codeigniter dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu:

1. *Model*, kode program (berupa OOP *class*) yang digunakan untuk memanipulasi *database*.
2. *View*, berupa template html/xml atau php untuk menampilkan data pada browser.
3. *Controller*, kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol *model* dan view).

#### 2.4.5 Leaflet

Leaflet adalah JavaScript Library yang berifat opensource dalam membangun peta interaktif yang Mobile friendly. Leaflet js memberikan



kemudahan dalam pengaplikasian kode sumber ke berbagai jenis peta digital dalam bentuk web. Karena leaflet bersifat open source, komunitas pengembang telah banyak berkontribusi plugin gratis untuk menyelesaikan berbagai tugas pemetaan.[12]

## **2.5 Penelitian Terdahulu**

Berikut beberapa contoh penelitian yang sudah dilakukan oleh peneliti yang dapat digunakan sebagai acuan dan pengetahuan:



Tabel 2.7 Referensi Penelitian

<b>Jurnal Pemanding</b>	1 (satu)
<b>Nama Jurnal</b>	Jurnal Teknologi dan Manejemen <i>Informatika</i>
<b>Judul</b>	Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta[7]
<b>Penerbit dan Tahun Terbit</b>	<a href="http://jurnal.unmer.ac.id/index.php/jtmi">http://jurnal.unmer.ac.id/index.php/jtmi</a> , Tahun 2020
<b>Penulis</b>	Aldy Cantona
<b>Permasalahan</b>	Pada penelitian ini, algoritma Dijkstra digunakan untuk mengkalkulasi jarak paling dekat dari satu titik ke museum yang dipilih menjadi tujuan. Pada aplikasi ini dirancang agar beroptimasi mencari rute ke museum dengan bobot jarak paling pendek di kota Jakarta. Dalam proses pencarian mampu diselesaikan menggunakan metode algoritma dijkstra sehingga kita mampu mengetahui tahapan kerja aplikasi dengan solusi perhitungan metode algoritma dijkstra.
<b>Tujuan Penelitian</b>	Mencari jalur terpendek dengan metode algoritma dijkstra dalam bentuk graf dengan bobot dan pemetaan area yang saling terhubung melalui jalur yang sudah ditentukan. Aplikasi ini dirancang agar berjalan pada smartphone dengan platform android yang digunakan mayoritas pada saat ini.
<b>Metode Penelitian</b>	Metode <i>Research and Development</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Aplikasi yang dibangun hanya memiliki satu titik <i>point</i> tujuan untuk dituju yaitu Museum-Museum di Jakarta. Kelebihan:



	Penulis menggunakan <i>Flutter</i> untuk membangun aplikasi sehingga ramah dan cocok untuk pengguna <i>smartphone</i> .
--	---

<b>Jurnal Pemanding</b>	2 (dua)
<b>Nama Jurnal</b>	<i>Riemann Research of Mathematics and Mathematics Education</i>
<b>Judul</b>	IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE TERPENDEK TEMPAT WISATA DI KABUPATEN KLATEN[13]
<b>Penerbit dan Tahun Terbit</b>	<i>Riemann Research of Mathematics and Mathematics Education</i> Volume 2, No. 1, April 2020, hal. 1-9
<b>Penulis</b>	Nugroho Arif Sudiby, Permadi Eka Setyawan, Yohana Putra Surya, Rahmad Hidayat
<b>Permasalahan</b>	Dalam pengambilan jalur untuk memudahkan bagi wisatawan lokal maupun mancanegara, ini sangat diperlukan untuk mencari jalur tercepat menuju salah satu titik wisata air.
<b>Tujuan Penelitian</b>	Dalam artikel ini akan dibahas bagaimana penerapan algoritma Dijkstra dalam mencari rute terpendek pada pencarian objek wisata di Kabupaten Klaten dengan bantuan program Tora.
<b>Metode Penelitian</b>	Metode <i>Dijkstra</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Pada aplikasi yang dibangun penulis hanya menunjukkan titik-titik yang harus dilalui berupa huruf sebagai penanda tempat wisata yang telah dimasukkan sebelumnya, dan hanya menunjukkan jarak yang dibutuhkan secara teks saja, tidak memvisualisasikan rute untuk <i>user</i> .



	Kelebihan: Pada penelitian ini, penulis menguraikan dengan lengkap penerapan algoritma <i>Dijkstra</i> secara matematis.
--	---

<b>Jurnal Pemanding</b>	3 (tiga)
<b>Nama Jurnal</b>	<i>Journal of Informatics and Computer Science</i>
<b>Judul</b>	Implementasi Sistem Informasi Geografis Jalur Pendakian Gunung Penanggungan Dengan Metode Dijkstra Dan Penerapan Fuzzy Dalam Rekomendasi Jalur[14]
<b>Penerbit dan Tahun Terbit</b>	JINACS: Volume 04 Nomor 03, 2023
<b>Penulis</b>	Martha Yogi Yuda Rifendy, Paramitha Nerisafitra
<b>Permasalahan</b>	Kegiatan mendaki gunung adalah kegiatan yang relatif tinggi tingkat bahayanya yang mengharuskan pendaki berjalan di hutan dan menghabiskan waktu yang cukup lama dengan kadar oksigen yang semakin menipis serta suhu yang sangat dingin bahkan bisa mencapai dibawah 0 derajat Celcius. Gunung Penanggungan merupakan objek untuk melakukan kegiatan pendakian bagi para pendaki. Untuk mendaki ke Gunung Penanggungan para pendaki bisa melalui beberapa jalur resmi pendakian yaitu Pendakian via Tamiajeng, Kedungudi, dan Jolotundo, namun kebanyakan para pendaki memilih jalur pendakian via Tamiajeng karena memiliki aksesibilitas yang tinggi dan jarak tempuh yang relatif singkat. Dari data tim SAR Gunung Penanggungan via Tamiajeng dari tahun 2015 - 2019 tercatat sebanyak 391 pendaki gunung mengalami kecelakaan saat pendakian, hal itu disebabkan oleh beberapa faktor yaitu



	tentang kurangnya pengetahuan gunung yang akan didaki, kurang mempersiapkan logistik yang diperlukan, dan jalur yang tidak sesuai dengan kemampuan pendaki.
<b>Tujuan Penelitian</b>	Tujuan dari penelitian ini akan membuat sebuah sistem informasi geografis berbasis website pendakian Gunung Penanggungan, yang mana website tersebut akan memakai Algoritma Dijkstra yang dipakai dalam memecahkan permasalahan jalur terpendek dari dengan menghitung cost (jarak) yang ada dan juga perhitungan Fuzzy Tahani untuk menentukan jalur rekomendasi pendakian terbaik sesuai dengan kriteria yang ditentukan.
<b>Metode Penelitian</b>	Metode <i>Dijkstra</i> dan Metode <i>Fuzzy</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Pada penelitian ini, penulis memanfaatkan metode <i>Fuzzy Tahani</i> untuk membiarkan <i>user</i> memilih jalur mana yang dilewati sesuai dengan keinginan pendaki apakah ingin medan yang curam, normal atau landau, ingin melewati jalur yang terdapat candi atau tidak.  Kelebihan: Pada penelitian ini, penulis memanfaatkan fungsi pencarian rute ini untuk komersil dengan membuat halaman <i>booking</i> untuk mempersiapkan bahan dan alat yang dipakai untuk rute yang dipilih pendaki.
<b>Jurnal Pemanding</b>	4 (empat)
<b>Nama Jurnal</b>	<i>JOURNAL OF MANAGEMENT Small and Medium Enterprises (SME's)</i>



<b>Judul</b>	ANALISIS PERBANDINGAN METODE MANAJEMEN PROYEK TI YANG PALING SERING DIGUNAKAN DI INDONESIA DAN LUAR NEGERI: A LITERATURE REVIEW[5]
<b>Penerbit dan Tahun Terbit</b>	Small and Medium Entreprises (SME's), Vol 15, No. 2, July 2022, p299-316
<b>Penulis</b>	Afifa Witania, Ardina Dana Nugraha, Ermawati, Luluk Fajar Sari, Nikita Lia, Megawati, Nuzul Nur Fadillah
<b>Permasalahan</b>	Berdasarkan sebuah studi, sebesar 90% professional IT percaya bahwa metode manajemen proyek TI memengaruhi tingkat keberhasilan proyek (Terlizzi et al., 2016). Hal tersebut menunjukkan pentingnya pemilihan metode manajemen proyek TI yang tepat. Metode manajemen proyek TI yang sering digunakan adalah agile (Marnada et al., 2021. Kharisma & Santoso, 2020. Hendrawan & Santoso, 2020) dan waterfall (Septiani, 2018. Shandy, 2019). Selain agile dan waterfall, metode <i>kanban</i> dan <i>scrum</i> juga dapat diterapkan oleh perusahaan.
<b>Tujuan Penelitian</b>	Tujuan dalam penelitian ini adalah untuk mengetahui metode yang paling sering digunakan oleh perusahaan dalam tata kelola manajemen proyek TI baik perusahaan dalam negeri maupun luar negeri. Selain itu, paper ini juga bertujuan untuk mengetahui metode mana yang paling efektif digunakan.
<b>Metode Penelitian</b>	<i>Systematic Literature Review</i> (SLR)
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Penulis menjelaskan bahwa metode <i>agile</i> lebih populer digunakan di luar negeri. Kelebihan: Penulis menguraikan berbagai kelebihan dan kekurangan antar metode pengembangan sistem.



<b>Jurnal Pemanding</b>	5 (lima)
<b>Nama Jurnal</b>	Jurnal Teknik
<b>Judul</b>	Rancangan Sistem E-itinerary Wisata Di Kawasan Jakarta Utara Dengan Penerapan Algoritma Dijkstra[15]
<b>Penerbit dan Tahun Terbit</b>	JT: Jurnal Teknik Vol. 12 No. 01 Th. 2023
<b>Penulis</b>	Destriana Widyaningrum, Chyquitha Danuputri
<b>Permasalahan</b>	Jarak antara lokasi wisata satu dengan lainnya pun tidak terlalu jauh, namun luasnya daerah wisata di kawasan tersebut tidak cukup di kunjungi hanya dalam satu hari. Bagi wisatawan luar kota maupun luar negeri akan kesulitan menentukan waktu yang efektif dalam menentukan wisata mana yang akan dikunjungi terlebih dahulu.
<b>Tujuan Penelitian</b>	Dalam mendukung sektor pariwisata, akan dirancang sistem yang dapat memberikan rekomendasi rencana perjalanan wisata dengan mengurutkan jarak paling terdekat antar lokasi wisata, untuk mengoptimalkan waktu kunjungan di beberapa tempat wisata di Jakarta Utara.
<b>Metode Penelitian</b>	Metode <i>waterfall</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Metode pengembangan yang digunakan adalah <i>waterfall</i> , pengembangan yang bertahap memakan waktu sedikit lebih lama. Kelebihan: Aplikasi dapat menyarankan berbagai rute yang dapat diambil oleh pengguna.



<b>Jurnal Pemanding</b>	6 (enam)
<b>Nama Jurnal</b>	<i>Building of Informatics, Technology and Science (BITS)</i>
<b>Judul</b>	Penerapan Metode Dijkstra Pada Jalur Distribusi LPG Untuk Penentuan Jarak Terpendek[16]
<b>Penerbit dan Tahun Terbit</b>	<i>Building of Informatics, Technology and Science (BITS)</i> Volume 3, No 3, December 2021, Page 235–243
<b>Penulis</b>	Novi Hendri Adi, Muhammad Giatman, Wakhinuddin Simatupang, Afrinal, Ronal Watrianthos
<b>Permasalahan</b>	Penentuan sesuatu jalur terpendek adalah hal yang krusial serta diperlukan sehubungan menggunakan optimasi waktu yang dipergunakan dan beberapa penghematan dibidang lainnya. Penentuan jalur terpendek memiliki alasan untuk menghemat bahan bakar, waktu dan tenaga yang diberikan.
<b>Tujuan Penelitian</b>	Penelitian ini bertujuan merancang software penentuan jarak terpendek pada jalur distribusi gas LPG berbasis web dengan menerapkan prosedur pemecahan Dijkstra pada PT. Amartha Anugrah Mandiri.
<b>Metode Penelitian</b>	Metode <i>Waterfall</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Pada aplikasi yang dibangun rute yang dipilih akan menampilkan arahan seperti belok, lurus perjalanan. Kelebihan: Memudahkan operatur truk pengangkut gas LPG dalam melakukan operasi dari satu pangkalan ke pangkalan lainnya.



<b>Jurnal Pemanding</b>	7 (tujuh)
<b>Nama Jurnal</b>	SMATIKA Jurnal
<b>Judul</b>	Penentuan Jarak Terpendek Menggunakan Metode Dijkstra Pada Data Spasial Openstreetmap (Studi Kasus : Pada Perusahaan Pengantaran Barang Wahana Logistik Kota Malang)[17]
<b>Penerbit dan Tahun Terbit</b>	SMATIKA Jurnal Volume 09 Nomor 01, April Tahun 2019
<b>Penulis</b>	Yan Watequlis Syaifudin, Dwi Puspitasari, Muhammad Ashari Fajar Nugroho
<b>Permasalahan</b>	Banyak jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan legalitas. Hal ini membuat user tidak dapat menggunakan peta tersebut secara bebas. Dan juga pada openstreetmap masih belum mendukung untuk pencarian jalur dengan input data lebih dari 1 tujuan.
<b>Tujuan Penelitian</b>	Wahana logistik merupakan salah satu jasa pengiriman logistik yang ada di kota Malang. Namun pada <i>openstreetmap</i> , lokasi wahana logistik masih belum ada sama sekali. Maka dari itu dibuatlah sebuah sistem penentuan jalur terpendek menggunakan algoritma <i>dijkstra</i> pada studi kasus wahana logistik.
<b>Metode Penelitian</b>	<i>Haversin formula</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Cara kerja sistem tersebut adalah memprioritaskan tujuan yang dipilih oleh user, kemudian untuk menentukan jalur selanjutnya akan dimulai dari lokasi yang sudah ditentukan sebelumnya, dan proses tersebut akan berkelanjutan pada pencarian jalur menuju titik tujuan user sesuai dengan prioritas yang diinginkan user.



<b>Jurnal Pemanding</b>	8 (delapan)
<b>Nama Jurnal</b>	Jurnal Sains Komputer & Informatika (J-SAKTI)
<b>Judul</b>	Penerapan Metode Dijkstra Pada Sistem Informasi Pencarian Jarak Terpendek Menuju Rumah Sakit di Wilayah Jakarta Barat[18]
<b>Penerbit dan Tahun Terbit</b>	Jurnal Sains Komputer & Informatika (J-SAKTI) Volume 5 Nomor 2, September 2021, pp. 540-550
<b>Penulis</b>	Imam Syarifudin, Yunita, Kresna Ramanda
<b>Permasalahan</b>	Pusat kesehatan memiliki peranan yang sangat penting karena dapat membantu mengobati dan merawat orang yang sakit serta memberikan pelayanan unit gawat darurat (UGD). Bagi warga pendatang di wilayah Jakarta Barat yang masih belum mengetahui dimana letak pusat kesehatan dan harus melewati jalur mana yang lebih dekat dari tempat mereka berada.
<b>Tujuan Penelitian</b>	Terciptanya rute terpendek menggunakan Metode <i>Dijkstra</i> untuk membantu memberikan informasi kepada masyarakat. Waktu untuk mencari pusat kesehatan menjadi lebih singkat dan rute menjadi lebih dekat.
<b>Metode Penelitian</b>	Metode <i>Simple random</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Aplikasi yang dibangun hanya menampilkan rute berupa panduan berupa nama tempat ke tempat lainnya. Kelebihan:



	Peneliti mengumpulkan sampel dari berbagai rumah sakit termasuk yang berada di daerah pinggiran, sehingga rumah sakit yang berada di pinggiran juga dapat terjamah.
--	---

<b>Jurnal Pemanding</b>	9 (sembilan)
<b>Nama Jurnal</b>	Jurnal PETIK
<b>Judul</b>	Penerapan <i>Kanban</i> Model Sebagai Metode Perancangan Sistem Informasi (Studi Kasus: Pemetaan Sekolah SMA/K/MA Kota Tasikmalaya)[6]
<b>Penerbit dan Tahun Terbit</b>	Jurnal PETIK Volume 8, No 1, Maret 2022
<b>Penulis</b>	Rudi Hartono
<b>Permasalahan</b>	Metode <i>kanban</i> baru-baru ini mendapatkan perhatian dan popularitas yang meningkat, karena banyak keuntungan yang membuatnya berkinerja lebih baik daripada metode lain dalam hal mengelola proyek perangkat lunak. Selain itu, ia memiliki papan yang digunakan untuk memvisualisasikan alur kerja dan memantau kemajuan proyek. Walaupun Konsep <i>kanban</i> pertama kali digunakan dalam sistem manufaktur Toyota[2] metode ini juga dapat diterapkan pada perancangan sistem informasi.
<b>Tujuan Penelitian</b>	Tujuan dari penelitian ini adalah untuk menerapkan pemodelan <i>kanban</i> dalam perancangan sistem informasi, karena studi kasus diimplementasikan pada sistem informasi berbasis GIS untuk pemetaan SMA atau SMK di Tasikmalaya
<b>Metode Penelitian</b>	Metode Agile <i>Kanban</i>



<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	<p>Perbedaan:</p> <p>Pada penelitian ini, penulis tidak menampilkan rute secara visual, hanya berupa lokasi saja.</p> <p>Kelebihan:</p> <p>Penelitian ini menggunakan <i>API Google Maps</i>, sehingga bila ingin mencari rute ke sekolah tertentu bisa dilakukan dengan menekan nama sekolah dan akan langsung <i>direct</i> ke <i>Google Maps</i>.</p>
---	--

<b>Jurnal Pemanding</b>	10 (sepuluh)
<b>Nama Jurnal</b>	JURNAL MEDIA INFORMATIKA BUDIDARMA
<b>Judul</b>	E-Travel Riau Berbasis Mobile Menggunakan Metode Dijkstra[19]
<b>Penerbit dan Tahun Terbit</b>	JURNAL MEDIA INFORMATIKA BUDIDARMA Volume 5, Nomor 4, Oktober 2021, Page 1336-1344
<b>Penulis</b>	Prina Marni, Hadi Asnal, Susi Erlinda, Agustin
<b>Permasalahan</b>	<p>Travel merupakan salah satu transportasi yang sering digunakan oleh study tour atau pariwisata di Provinsi Riau.</p> <p>Travel tidak hanya bergerak dalam bidang pemesanan tetapi juga di bidang pengiriman barang. Tetapi yang sering terjadi pada travel yaitu pemesanan yang masih dilakukan secara manual. Proses Pengorderan dilakukan melalui telephon, lalu admin mencatatat alamat penumpang yang akan dijemput dan admin langsung mengkonfirmasi ke driver untuk menjemput penumpang.</p>
<b>Tujuan Penelitian</b>	Tujuan dari penelitian ini adalah untuk membantu penumpang dalam pemesan travel secara online dan driver



	dapat memantau dan menentukan lokasi calon penumpang. Pada penelitian ini dibuat sebuah aplikasi pemesanan travel online berbasis android menggunakan algoritma Dijkstra.
<b>Metode Penelitian</b>	Metode <i>Dijkstra</i>
<b>Perbedaan dan Kelebihan Penelitian Penulis</b>	Perbedaan: Pada penelitian ini, penulis menggunakan penambahan jadwal untuk memulai perutean dengan input <i>form</i> Kelebihan: Aplikasi ini dapat menyimpan transaksi atas perjalanan travel yang dilakukan, <i>driver</i> dapat mengetahui detail penting dari jadwal seperti nama penumpang, dan rute yang didukung <i>google maps</i> .

