



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

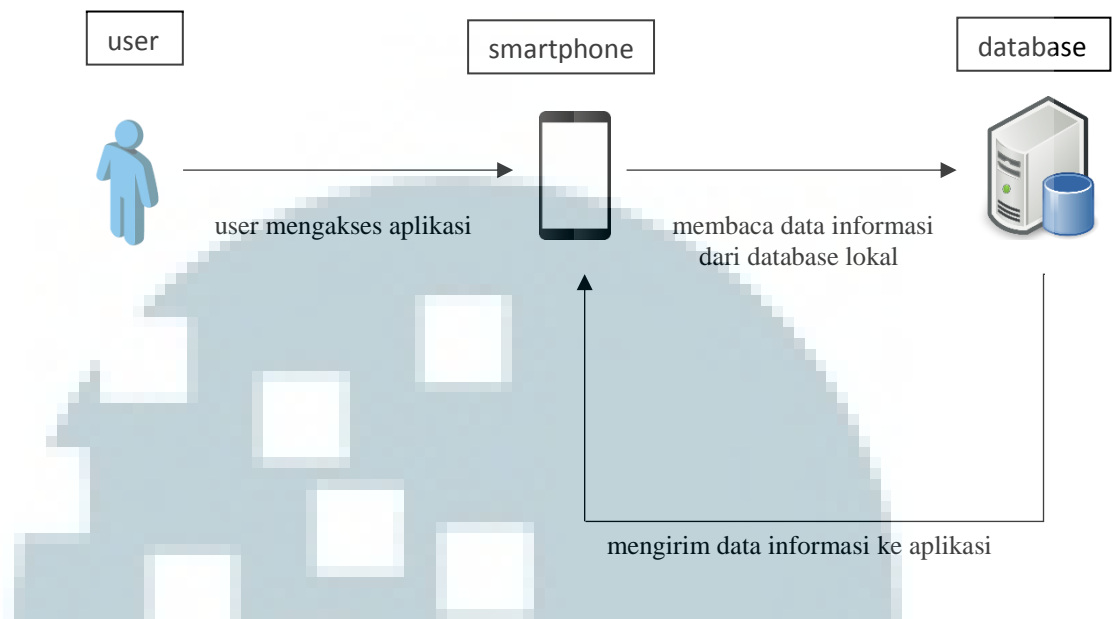
BAB III

METODOLOGI PENELITIAN

3.1. Gambaran Umum Obyek Penelitian

Jakarta merupakan tempat berdirinya kantor-kantor pusat BUMN, perusahaan swasta, dan perusahaan asing. Menurut Dinas Perhubungan DKI (2016), tercatat 46 kawasan dengan 100 titik simpang rawan macet di Jakarta. Definisi rawan macet adalah arus tidak stabil, kecepatan rendah serta antrean panjang. Untuk melayani mobilitas penduduk Jakarta, pemerintah menyediakan berbagai macam sarana transportasi umum. Salah satunya adalah bus kota yang dikelola oleh pihak swasta, seperti Mayasari Bhakti, Metro Mini, Kopaja, dan Bianglala. Bus-bus ini melayani rute yang menghubungkan halte yang ada di dalam kota.

Sistem yang akan dirancang diperuntukkan bagi para pengguna *smartphone* Android untuk mencari halte bus sesuai dengan trayeknya. Pengguna dapat menentukan lokasi yang ingin dituju kemudian sistem akan membantu dalam pencarian halte bus. Aplikasi akan menunjukkan rute terpendek menggunakan algoritma Dijkstra, dan jenis bus yang harus dinaiki oleh pengguna.



Gambar 3.1 Gambaran Umum Sistem

Berdasarkan gambar 3.1 pengguna diharuskan mengakses aplikasi melalui *smartphone* dan kemudian pengguna tersebut melakukan *request* terhadap aplikasi. Selanjutnya aplikasi membaca data informasi dari *database* sesuai dengan permintaan *user*. *Database* mengirimkan informasi dan data yang diminta sebelumnya melalui *smartphone*, hingga pada akhirnya pengguna mendapatkan informasi yang diinginkan.



3.2. Penelitian Sebelumnya

Dalam melakukan penelitian dibutuhkan referensi-referensi untuk membantu memperkuat pernyataan terhadap masalah yang diteliti. Berikut adalah tabel dari beberapa jurnal ilmiah yang digunakan dalam penelitian ini :

Tabel 3.1. Penelitian Sebelumnya

1	Pengarang	Dibi Khairurrazi Budiarsyah
	Judul	Algoritma Dijkstra, Bellman-Ford dan Floyd-Warshall untuk Mencari Rute Terpendek dari Suatu Graf
	Metode	Algoritma Dijkstra, Bellman-Ford dan Floyd-Warshall
	Hasil / Kesimpulan	Algoritma Dijkstra lebih cepat penggunaannya dibandingkan dengan algoritma Bellman-Ford dan Floyd-Warshall.
2	Pengarang	Rani Susanto, Anna Dara Andriana
	Judul	Perbandingan Model <i>Waterfall</i> dan <i>Prototyping</i> untuk Pengembangan Sistem Informasi
	Metode	<i>Prototyping</i> dan <i>Waterfall</i>
	Hasil / Kesimpulan	Metode <i>prototype</i> cocok untuk membuat perangkat lunak yang bisa di ubah sesuai dengan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai

		<p>untuk tugas akhir/skripsi yang memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus.</p>
--	--	---

Berdasarkan data penelitian sebelumnya yang terdapat pada tabel 3.1 yang menjelaskan bahwa algoritma Dijkstra lebih cepat penggunaannya dibandingkan algoritma Bellman-Ford dan Floyd Warshall, maka penulis menggunakan algoritma Dijkstra untuk diterapkan dalam aplikasi pencarian rute bus yang terpendek.

3.3. Variabel Penelitian

Berdasarkan penelitian yang dilakukan, terdapat variabel yang digunakan untuk diteliti. Variabel tersebut dibagi menjadi dua (2) jenis yaitu sebagai berikut:

a) Variabel bebas

Variabel bebas yang terdapat dalam penelitian ini adalah variabel x yang merupakan lokasi titik awal pengguna dan variabel y yang merupakan lokasi tujuan pengguna berdasarkan rute busnya.

b) Variabel terikat

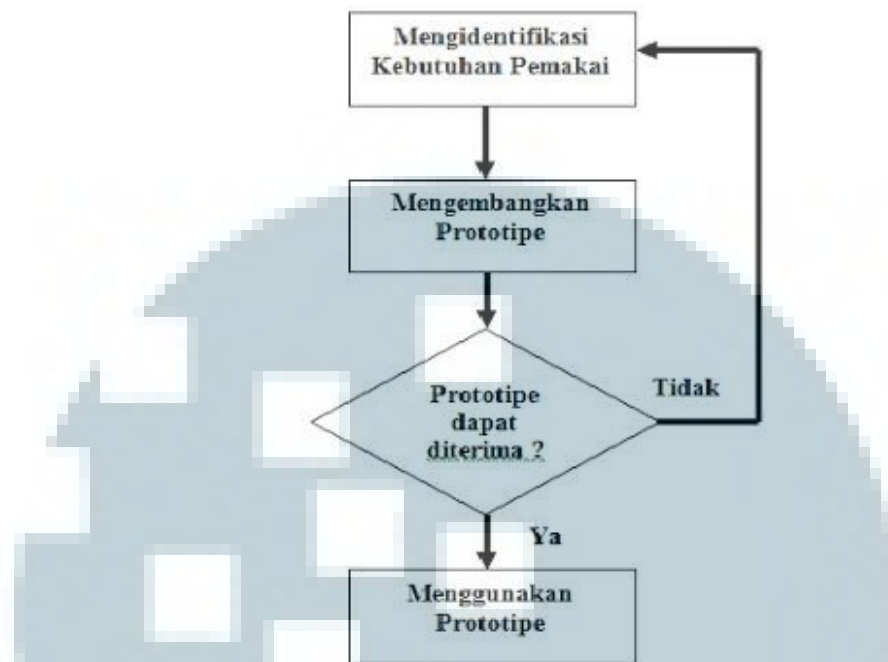
Variabel terikat yang digunakan dalam penelitian ini adalah variabel s yang merupakan jarak terdekat dari lokasi *user* ke halte bus berdasarkan lokasi destinasi dari penggunanya.

3.4. Metode Penelitian

Metode yang akan digunakan dalam perancangan aplikasi ini adalah menggunakan metode *prototype*. Metode *prototype* merupakan metode yang menyajikan gambaran sistem secara lengkap, pengguna dapat melihat pemodelan sistem dari sisi tampilan maupun teknik prosedural yang akan dibangun. Tujuan dari penggunaan metode *prototype* adalah untuk melakukan pengembangan model awal *software* menjadi sebuah *software* yang *final*.

Langkah pertama menggunakan metode *prototype* adalah dengan mengidentifikasi kebutuhan pemakai. Mengidentifikasi kebutuhan pemakai dilakukan dengan cara bertanya secara langsung kepada pemakai. Setelah diidentifikasi, tahap selanjutnya adalah dengan mengembangkan *prototype* tersebut. Jika *prototype* tersebut diterima dengan baik, maka *prototype* tersebut dapat digunakan untuk perancangan sistem pada tahap selanjutnya yaitu pada tahap *final*.

UMMN



Gambar 3.2 Metode *Prototype*

Berikut tabel perbandingan antara metode *prototype* dengan metode *waterfall* berdasarkan komponen – komponen yang ada sebagai berikut:

UMMN

Tabel 3.2. Perbandingan Metode *Prototype* dengan *Waterfall*

Komponen Perbandingan	<i>Prototype</i>	<i>Waterfall</i>
Waktu Pengerjaan	Lebih singkat, pengerjaan bisa langsung dilakukan bersamaan dengan analisa <i>feedback</i> dari <i>user</i>	Lebih lama, harus sesuai dengan tahapan-tahapan yang ada.
Cara pengerjaan	Pada saat pengerjaan dapat berubah sesuai dengan permintaan <i>user</i>	Pada saat pengerjaan tidak mengakomodasi perubahan secara langsung
Interaksi dengan <i>user</i>	Interaksi aktif	Interaksi pasif

Berdasarkan Tabel 3.2 dapat dianalisa bahwa dalam hal waktu pengerjaan metode prototipe dapat diselesaikan lebih cepat karena pengerjaan bisa dilakukan bersamaan dengan analisa langsung untuk kebutuhan *user*, sedangkan metode *waterfall* memiliki tahapan-tahapan yang lebih banyak sehingga waktu pengerjaan akan lebih lama karena harus menyesuaikan dengan tahapan-tahapan yang ada sebelum pengerjaan terlebih dahulu.

3.5. Penggunaan Algoritma Dijkstra

Dalam penelitian ini, penulis menggunakan algoritma Dijkstra untuk melakukan pencarian rute terpendek menuju halte bus berdasarkan trayek busnya. Perbedaannya dengan algoritma Google Maps adalah algoritma Dijkstra melakukan pencarian terhadap semua titik kemudian memilih nilai yang dianggap paling pendek dari setiap titik dan pada akhirnya hanya memberikan lintasan jalur terpendek, sedangkan pada algoritma Google Maps tidak dijelaskan tentang algoritma yang digunakan karena sistemnya yang lebih kompleks sehingga membuat Google Maps dapat memodifikasi algoritmanya sesuai dengan kebutuhan.

Algoritma Dijkstra dapat digunakan untuk mencari rute terpendek dari suatu grafik berbobot tanpa nilai negatif, waktu yang dibutuhkan untuk operasi ini adalah $O(|V|^2)$ dimana V adalah jumlah *node*. Algoritma Bellman-Ford fungsinya sama dengan Algoritma Dijkstra, hanya saja prosesnya memakan waktu yang lebih lama yakni $O(V.E)$ dengan E adalah jumlah sisi/*edge*. Meskipun memakan waktu lebih lama, Algoritma Bellman-Ford dapat menangani sisi berbobot negatif. Algoritma Floyd-Warshall juga dapat menangani kasus serupa, waktu yang dibutuhkan adalah $O(|V|^3)$. Untuk sisi tanpa bobot negatif, algoritma Dijkstra lebih cepat dalam menentukan rute terpendek, namun apabila ada sisi berbobot negatif, algoritma Bellman lebih disukai. Dari beberapa algoritma ini terdapat kesamaan, yakni semuanya tidak dapat menentukan rute terpendek dari graf bersiklus negatif. Aplikasi dari algoritma-algoritma ini adalah dalam

menentukan rute terpendek yang dapat diambil, misalnya saja rute perjalanan dari suatu kota ke kota yang lain (Dibi, 2010).

3.6. Penggunaan Database

Database yang digunakan dalam perancangan aplikasi ini adalah menggunakan *database* Sqlite. Data yang disimpan adalah data yang sudah berbentuk data koordinat jalur dan grafnya. Tabel yang akan dibuat pada *database* Sqlite adalah sebagai berikut:

a) Tabel angkutan_umum

Tabel angkutan_umum digunakan untuk menyimpan data-data bus seperti nama dan nomor trayek bus juga rute yang dilalui bus menggunakan kode simpul yang merupakan kumpulan dari titik koordinat. Kode simpul digunakan untuk membantu aplikasi dalam membaca rute trayek pada bus. Misalnya kode simpul (0-1) yang berarti bus tersebut berjalan dari lokasi 0 menuju lokasi 1.

b) Tabel graph

Tabel graph digunakan untuk menyimpan data jalur berdasarkan kode simpul menggunakan koordinat yang didapatkan dari Google Maps.

c) Tabel lokasi_tujuan

Tabel lokasi_tujuan digunakan untuk menyimpan data lokasi yang dituju menggunakan koordinat yang didapatkan dari Google Maps.

Tabel tersebut dipanggil oleh fungsi - fungsi yang dibuat untuk di tampilkan datanya dan memberikan informasi kepada *user* yaitu informasi lokasi dan rute bus yang dilalui untuk menuju halte tujuan tersebut.

3.7. Teknik Pengumpulan Data

a) Observasi

Teknik pengumpulan data dalam penelitian ini adalah menggunakan metode observasi. Observasi dalam penelitian ini tidak secara langsung dilakukan dilapangan melainkan melalui Google Maps. Penulis hanya mencari data lokasi melalui Google Maps karena penulis menggunakan API Google Maps untuk perancangan aplikasinya.

b) Studi Pustaka

Dalam penelitian ini penulis menggunakan metode studi pustaka untuk mendapatkan referensi yang bisa dipertanggung jawabkan kebenarannya terhadap masalah yang diteliti. Adapun bahan pustaka yang digunakan adalah *website*, buku-buku referensi dan jurnal ilmiah.

3.8. Teknik Pengambilan Data

Teknik yang digunakan dalam pengambilan data adalah dengan menggunakan API (*Application Programming Interface*) Google Maps. API Google Maps menyediakan fungsi-fungsi dan fitur-fitur yang dibutuhkan dalam penelitian ini, seperti kode untuk mendapatkan koordinat

garis lintang dan garis bujur dari suatu lokasi dan fungsi untuk melakukan *tracking* menggunakan GPS (*Global Positioning System*).

3.9. Teknik Pengolahan Data

Teknik yang digunakan dalam pengolahan data adalah dengan menggunakan IDE (*Integrated Development Environment*) Android Studio. IDE sendiri merupakan sebuah *tools development* yang memiliki beberapa fitur yang diperlukan dalam pengembangan perangkat lunak. IDE Android Studio digunakan untuk melakukan perubahan pada *pseudocode* algoritma Dijkstra menjadi algoritma pemrograman dan membuat rancangan aplikasi Android. Alasan penulis menggunakan IDE Android Studio adalah karena Google sudah tidak mendukung pengembangan IDE Eclipse.

UMMN