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.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma Floyd-Warshall

2.1.1 Definisi

Algoritma Floyd-Warshall (dapat juga disebut WFI *algorithm* atau Roy-Floyd *algorithm*) merupakan algoritma *graph analysis* yang digunakan untuk menemukan jarak terpendek dari sebuah *graph* yang mempunyai *weight* dan arah. Algoritma ini akan mencari jarak terpendek dari semua *vertex* yang ada pada *graph*. Algoritma Floyd-Warshall ditemukan pada tahun 1962 dan dinamai berdasarkan nama kedua penemunya, yaitu Robert Floyd dan Stephen Warshall. Algoritma ini merupakan contoh dari *dynamic programming*, dimana data yang disimpan oleh program dapat berubah secara *real time* (Kiruthika dan Umarani, 2012). Dalam algoritma ini terdapat verteks *intermediate* untuk *path* terpendek. Verteks *intermediate* yang dimaksud adalah verteks yang bukan merupakan verteks pertama maupun verteks terakhir dari sebuah *path*. *Path* adalah "jalur" yang dilalui dari verteks awal ke verteks akhir. Sebuah *path* yang hanya mengandung satu *edge* tidak akan mempunyai verteks *intermediate*. Misalkan verteks pada *graph* G adalah V = {1, 2, 3, ..., n}. Untuk semua pasangan verteks i,j $\in$ V, semua *path* dari verteks i menuju verteks j dianggap memiliki verteks *intermediate* yang terdapat dalam set {1, 2, ..., k}, dan memiliki aturan sebagai berikut.

1. Jika k bukan merupakan verteks *intermediate* dari *path* p, maka verteks *intermediate* dari *path* p berada pada set {1, 2, ..., k-1}, dan *path* terpendek dari

Rancang Bangun Aplikasi..., Reynaldo Kevin, FTI UMN, 2018

verteks i menuju verteks j dengan verteks *intermediate* {1, 2, ..., k-1} juga merupakan *path* terpendek dari verteks i menuju verteks j dengan verteks *intermediate* {1, 2, ..., k}.

2. Jika k merupakan verteks *intermediate* dari *path* p, maka p dipecah menjadi $i \rightarrow (p1)k \rightarrow (p2)j$. p1 merupakan *path* terpendek dari verteks i menuju verteks k dengan verteks *intermediate* {1, 2, ..., k-1} dan p2 merupakan *path* terpendek dari verteks k menuju verteks j dengan verteks *intermediate* {1, 2, ..., k-1} (Chaudhary dkk., 2000).

Misalkan $d_{ij}^{(k)}$ merupakan *weight path* terpendek dari verteks i menuju verteks j dengan verteks *intermediate* {1, 2, ..., k}. Jika k = 0, *path* dari verteks i menuju verteks j yang tidak mempunyai verteks *intermediate* lebih besar dari 0 tidak mempunyai verteks *intermediate* sama sekali. *Path* tersebut mempunyai paling banyak satu *edge*, dan karena $d_{ij}^{(0)} = w_{ij}$, maka fungsi rekursifnya dapat didefinisikan menjadi rumus 2.1 berikut (Chaudhary dkk., 2000).

$$d_{ij}^{(k)} \begin{cases} w_{ij} & , jika\ k = 0 \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & , jika\ k \geq 1 \end{cases} \quad ...(2.1)$$

2.1.2 Cara Kerja

Menurut Magzhan dan Jani (2013), cara kerja algoritma Floyd-Warshall dijabarkan sebagai berikut.

1. Diberikan *graph* G yang mempunyai *node* sejumlah n.
2. Bentuk matriks n x n dan diisi dengan *weight* yang ada pada masing-masing *edge*.
3. *Node* awal ditandai dengan i, *node* akhir ditandai dengan j, dan *intermediate node*/*vertex* ditandai dengan k.

4. Isi matriks [i][i] (jarak *node* dengan *node* itu sendiri) dengan 0 dan matriks [i][j] dengan nilai *infinite* jika tidak ada *edge* yang menghubungkan, dengan catatan nilai *infinite* yang ada tidak boleh melebihi nilai maksimum tipe data dibagi 2 (MAX/2) agar tidak terjadi *overflow* saat proses perhitungan.

5. Lakukan iterasi sebanyak n^3, dan data akan di-*update* setiap ditemukan *path* yang lebih pendek.

Proses iterasi pada algoritma Floyd-Warshall ditunjukkan pada Gambar 2.1.

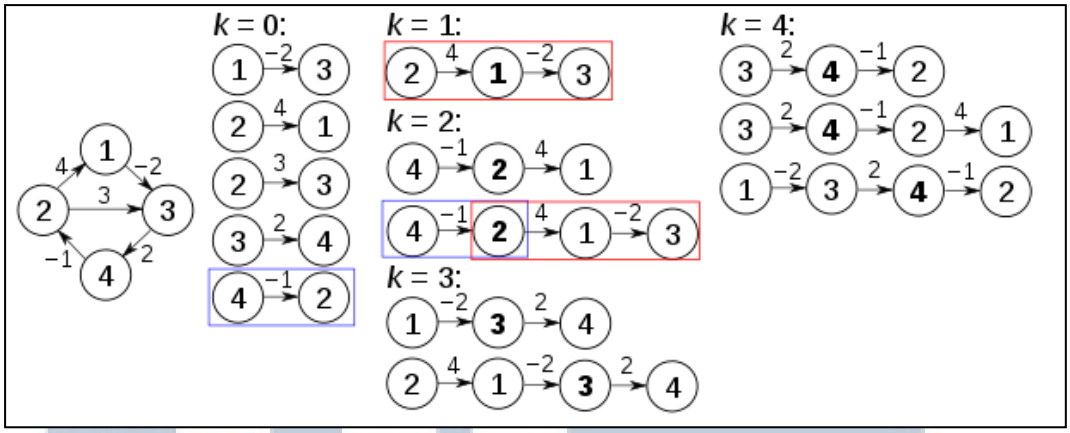
```
for (int k = 0; k < n; k++) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (d[i][j] > d[i][k] + d[k][j]) {

                /*
                 * d[i][j] -  is equal to
                 * the shortest path from i-th to j-th vertices.
                 */
                d[i][j] = d[i][k] + d[k][j];
            }
        }
    }
}
```

Gambar 2.1 Proses Iterasi Algoritma Floyd-Warshall
(Magzhan dan Jani, 2013)

Contoh dari algoritma Floyd-Warshall ditunjukkan pada Gambar 2.2.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 2.2 Contoh Algoritma Floyd-Warshall
(Chaudhary dkk., 2000)

Pada iterasi pertama (k = 0), *path* yang diketahui hanyalah path dengan *edge* tunggal pada *graph*. Pada saat k = 1, ditemukan *path* yang melalui verteks 1, yaitu *path* [2, 1, 3]. Karena *weight* dari *path* [2, 1, 3] lebih kecil dari *path* [2, 3], maka *weight* dari *path* [2, 3] diganti dengan *weight* dari *path* [2, 1, 3] meskipun mempunyai *edge* yang lebih sedikit. Pada k = 2, *path* yang melewati verteks [1, 2] ditemukan. *Path* [4, 2, 3] tidak termasuk ke dalam *list* karena *path* [4, 2, 1, 3] mempunyai *weight* yang lebih kecil (dari iterasi k = 1). Pada k = 3, ditemukan *path* yang melewati verteks [1, 2, 3]. Dan pada k = 4, semua *path* terpendek telah ditemukan.

2.2 TransJakarta

TransJakarta adalah sebuah sistem transportasi *Bus Rapid Transit* (BRT) pertama di Asia Tenggara dan Selatan dengan jalur lintasan terpanjang di dunia (208km). Sistem BRT ini didesain berdasarkan sistem TransMilenio di Bogota, Kolombia. TransJakarta resmi beroperasi pada 1 Februari 2004. TransJakarta dioperasikan dengan menggunakan bus sebanyak 1.347 unit bus, terdiri dari bus tunggal dan bus gandeng (PT TransJakarta, 2016). Setiap koridor memiliki jenis bus yang berbeda.

Hingga tahun 2013, TransJakarta memiliki 12 koridor utama yang saling terintegrasi dan tersebar di seluruh wilayah Jakarta dengan 228 halte yang dilalui. Halte TransJakarta didesain berbeda dari halte angkutan umum lainnya. Ketinggian platform (lantai halte) yang diatur setinggi 110 cm dari permukaan jalan, menyesuaikan dengan tinggi pintu bus. Letak halte TransJakarta umumnya berada di tengah jalan, kecuali jalan satu arah dan jalan dengan area pembatas jalan yang minim. Di dalam halte disediakan tempat duduk, tempat sampah (bus tertentu), dan papan informasi mengenai rute TransJakarta ataupun lainnya (PT TransJakarta, 2016). Sistem tiket pada halte TransJakarta menggunakan kartu elektronik (*e-ticketing*), sebagai pengganti uang tunai. Operator koridor tidak menerbitkan kartu tersebut, melainkan menggunakan kartu prabayar yang dikeluarkan oleh bank. Pelanggan melakukan transaksi tiket dengan cara Tap In di *Barrier Gate* saat masuk ke dalam halte dan Tap Out saat keluar dari halte. Saldo uang elektronik hanya akan terpotong pada saat Tap In (masuk ke halte). Tap out dilakukan sebagai tanda bahwa pelanggan telah sampai di lokasi tujuan. Tarif dari TransJakarta dibagi berdasarkan waktu pembelian tiketnya. Terdapat tiga *range* harga tiket Transjakarta, yaitu Rp2.000,- pada pukul 05.00 WIB-07.00 WIB, Rp3.500,- pada pukul 07.00 WIB-24.00 WIB, dan Rp3.500,- pada pukul 24.00 WIB-05.00 WIB (PT Transjakarta, 2016).

2.3 KRL Jabodetabek

KRL atau kereta rel listrik (sekarang disebut sebagai kereta *Commuter Line*) Jabodetabek adalah layanan kereta api yang beroperasi di wilayah Jabodetabek. Awalnya berada di bawah naungan PT KAI (Kereta Api Indonesia),

KRL jabodetabek kini berada di bawah naungan PT KAI Commuter Jabodetabek (PT KCJ) sejak 15 September 2008. Pada Oktober 2016, KRL Jabodetabek memiliki 826 unit KRL, melayani 72 stasiun di seluruh Jabodetabek dengan jangkauan rute mencapai 184,5 km dengan rata-rata jumlah pengguna per hari mencapai 850.000 pengguna pada hari-hari kerja dan rekor jumlah pengguna terbanyak yang dilayani dalam satu hari adalah 931.082 pengguna (PT KCJ, 2016).

Elektrifikasi jalur kereta api sudah ada sejak jaman penjajahan Belanda. Elektrifikasi pertama dilakukan untuk jalur lintas Tanjungpriok-Meester Cornelis (Jatinegara) pada tahun 1923 dan selesai pada 24 Desember 1924. Elektrifikasi kemudian terus berlanjut dengan mengoperasikan jalur lintas Batavia (Jakarta Kota)-Kemayoran, dan Meester Cornelis (Jatinegara)-Manggarai-Koningsplein (Gambir)-Batavia (Jakarta Kota). Sejak 1 Mei 1927, di Kota Batavia melintas KRL yang mengelilingi kota (*ceintuur-baan*). Tahun 1930, untuk pertama kalinya jalur KRL Batavia (Jakarta Kota)-Buitenzorg (Bogor) beroperasi. Hingga tahun 1939, telah ada sebanyak 72 perjalanan KRL melintasi jalur lingkar Batavia dan Manggarai-Bogor. Setelah Indonesia merdeka, lokomotif listrik masih beroperasi di sekitar Jakarta. Namun akhirnya usia kereta yang telah mencapai setengah abad, dan tidak ada penambahan lokomotif listrik baru, membuat transportasi dengan lokomotif listrik tidak lagi memadai. Perkeretaapian Jabodetabek kemudian mulai akrab dengan rangkaian KRL buatan Jepang yang mulai beroperasi tahun 1976 (PT KCJ, 2016).

2.4 Usability

Usability (kegunaan) perangkat *mobile* dan aplikasi yang ada di dalamnya berbeda dengan sistem komputer yang lain, dikarenakan karakteristik yang berbeda. Aplikasi pada perangkat *mobile* membutuhkan *handphone* ataupun tablet untuk dapat dijalankan karena aplikasi terpasang dalam *handphone* pada saat pengembangan maupun saat di*install* oleh *user*. *User* cenderung memilih aplikasi yang mudah dipelajari (*easy to learn*), lebih cepat dalam mengerjakan sebuah *task* (*efficient to use*), dan lebih *user-friendly* serta dapat memenuhi ekspektasi *user* (*user satisfaction*) (Nayebi dkk., 2012). ISO 9241 mendefinisikan *usability* sebagai suatu tingkat dimana sebuah produk dapat digunakan oleh *user* tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks kegunaan yang spesifik. Selain itu, ISO 25010 juga membagi kepuasan (*satisfaction*) menjadi *likeability* (kepuasan dalam tujuan pragmatis), *pleasure* (kepuasan dalam tujuan hedonis), *comfort* (kepuasan fisik), dan *trust* (kepuasan dalam keamanan) (Nayebi dkk, 2012).

Dalam melakukan evaluasi *usability* perangkat *mobile*, terdapat tiga tipe metodologi yang digunakan, yaitu *laboratory experiments*, *field studies*, dan *Hands-on Measurement*. *Laboratory experiments* merupakan tipe metodologi dimana partisipan diminta untuk melakukan *task* tertentu menggunakan aplikasi *mobile* di dalam sebuah lokasi yang dikontrol. *Field studies* merupakan tipe metodologi dimana *user* diminta untuk menggunakan aplikasi *mobile* yang sudah disediakan dan akan dievaluasi berdasarkan pengalaman *user* saat menggunakan aplikasi. *Hands-on measurement* merupakan tipe metodologi dimana aspek-aspek

tertentu dalam aplikasi *mobile* diuji secara langsung untuk dievaluasi kegunaannya (Nayebi dkk., 2012).

2.5 Skala Likert

Skala Likert merupakan skala pengukuran yang digunakan untuk mengukur sifat-sifat individu seseorang. Skala Likert pertama kali dikembangkan oleh Likert pada tahun 1932 dan mempunyai 5 (lima) poin titik respon yaitu sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju. Seiring dengan perkembangannya, peneliti mulai mencoba untuk membuat skala Likert dengan titik respon yang lebih banyak, di antaranya 7, 11, dan 15. Skala Likert termasuk ke dalam golongan skala interval, artinya setiap nilai pada titik respon skala Likert memiliki interval yang sama (Budiaji, 2013). Dalam sebuah pertanyaan yang mempunyai lima buah titik respon yang menggunakan skala Likert, masing-masing bernilai 1, 2, 3, 4, dan 5 (1 untuk sangat tidak setuju, 3 untuk netral, dan 5 untuk sangat setuju). Nilai maksimum dari skala Likert dengan lima buah titik respon adalah 5n, dan nilai minimumnya adalah n, dimana n adalah jumlah pertanyaan (Budiaji, 2013). Perhitungan persentase nilai akhir dari skala Likert menggunakan rumus 2.2 berikut.

$$\%skor = \frac{skor\ aktual}{skor\ ideal}\ x\ 100\% \qquad ...(2.2)$$

Skor aktual didapat dari jumlah pertanyaan dikalikan dengan nilai dari titik respon setiap pertanyaan yang dipilih responden sedangkan skor ideal didapat dari jumlah pertanyaan dikalikan dengan skor tertinggi pada setiap pertanyaan. Jika persentase nilai akhir berkisar antara 20%-36%, maka responden dianggap sangat tidak setuju. Jika persentase nilai akhir berkisar antara 36.1%-52%, maka responden dianggap tidak setuju. Jika persentase nilai akhir berkisar antara

52.1%-68%, maka responden dianggap netral. Jika persentase nilai akhir berkisar antara 68.1%-84%, maka responden dianggap setuju. Jika persentase nilai akhir berkisar antara 84.1%-100%, maka responden dianggap sangat setuju (Pakar, 2003).

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA