# IMPLEMENTASI ALGORITMA DIJKSTRA SEBAGAI SOLUSI EFEKTIF PEMBUATAN SISTEM BANTUAN BENCANA REAL TIME

**Siti Nandiroh[1], Haryanto[2] dan Hafidh Munawir[3]**

**Abstract**: Research on communications technologies that support by Dijkstra algorithm, as a determinant of the distribution are more effective for the volunteer. The system also equipped by the data inventory and needs of each refugee, so there is no buildup and lack support in their respective refugee camps, as well as appropriate aid, according to the type of the needs of victims. This study is a system which has the purpose of making supplies, disaster relief distribution channel based on the criteria of value chain, and the level of priority areas or disaster, and disaster guide information is up to date. Information systems that are formed based on the Djikstra algorithm, while the technology which is the basis of the system is mobile. The final results will be verified by BNPB and MDMC as well as local government and police disaster area. So the information system that will form the basis for appropriate and relevant to distributed.

**Keywords***: Djikstra algorithm, refugee, disaster, route, system.*

## PENDAHULUAN

Perlindungan masyarakat dari resiko ancaman bencana dapat tercapai apabila salah satunya yaitu informasi mengenai bencana yang akurat dari sumber yang terpercaya disampaikan secara cepat dan tepat pada sasaran yang menbutuhkan. Sumber yang terpercaya sangat diperlukan untuk menghindari informasi yang menyesatkan masyarakat, penyampaian secara cepat dan tepat sasaran sangat diperlukan agar masyarakat lebih waspada dan ada waktu yang cukup untuk melakukan penyelamatan. Penyampaian informasi secara cepat dapat dilakukan jika menggunakan perangkat komunikasi yang canggih, perangkat komunikasi yang canggih saat ini diantaranya adalah telepon seluler. Penggunaan telepon seluler pada penelitian ini di dukung oleh algoritma Dijkstra.

Pada tahun 2000 sudah banyak algoritma mencari lintasan terpendek yang pernah ditulis. Salah satunya algoritma yang paling sesuai untuk pencarian solusi terhadap kasus yang memerlukan *graph* alternatif adalah Algorima Dijkstra (sesuai dengan nama penemunya). Algoritma Dijkstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algorima ini juga benar untuk graf tak berarah. (Renaldi, 2001)

Misalkan sebuah graf berbobot n buah simpul dinyatakan dengan matriks ketetanggaan M=[$m_{ij}$], yang dalam hal ini,

[1] Jurusan Teknik Industri, Fakultas Teknik, Universitas Muhammadiyah Surakarta
Jl. A Yani Tromol Pos I Pabelan, Surakarta
Email : stnandiroh@rocketmail.com

[2] Jurusan Sistem Komputer STMIK AUB Surakarta
Jl. MW Maramis No.29 Cenglik Surakarta

[3] Jurusan Teknik Industri, Fakultas Teknik, Universitas Muhammadiyah Surakarta
Jl. A Yani Tromol Pos I Pabelan, Surakarta

Naskah diterima: 23 Des 2013, direvisi: 5 Jan 2014, disetujui: 7 Jan 2014

$m_{ij}$ = bobot sisi(i,j) (pada graaf berarah $m_{ij}=m_{ji}$)
$m_{ij}$ = 0
$m_{ij}$ = ∞, jika tidak ada sisi dari simpul *i* ke simpul *j*

Selain matrik *M*, kita juga menggunakan larik S = [$s_i$] yang dalam hal ini,
$s_i$ = 1, Jika simpul i termasuk ke dalam lintasan terpendek.
$s_i$ = 0, Jika simpul i tidak termasuk ke dalam lintasan terpendek.

dan larik/ tabel D = [$d_i$] yang dalam hal ini,
$d_i$ = panjang lintasan dari simpul awal ke simpul i

Algoritma Lintasan terpendek Dijkstra (mencari lintasan terpendek dari simpul a ke semua simpul lain)

Langkah 0 (inisialisasi)
- inisialisasi $s_i = 0$ dan $d_i = m_{ai}$ untuk i = 1,2,...,*n*

Langkah 1:
- isi $s_a$ dengan 1 (karena simpul a adalah simpul asal lintasan terpendek, jadi sudah pasti terpilih)
- isi $d_a$ dengan ∞ (tidak ada lintasan terpendek dari simpul a ke a)

Langkah 2,3,...,n-1:
- cari *j* sedemikin sehingga $s_j = 0$ dan $d_j$ = min {$d_1$,$d_2$,...,$d_n$}
- isi $s_j$ = dengan 1
- perbarui $d_i$, untuk i = 1,2,3,...,n dengan: $d_i$ (baru) = min {$d_i$ (lama),$d_j + m_{ij}$ }

Dari graf yang disebutkan di peroleh matrik ketetanggaan *M* yang ditunjukkan pada tabel 1.

**Tabel 1. Matrix ketetanggaan**

| | j = 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| i = 1 | 0 | 50 | 10 | 40 | 45 | ∞ |
| 2 | ∞ | 0 | 15 | ∞ | 10 | ∞ |
| 3 | 20 | ∞ | 0 | 15 | ∞ | ∞ |
| 4 | ∞ | 20 | ∞ | 0 | 35 | ∞ |
| 5 | ∞ | ∞ | ∞ | 30 | 0 | ∞ |
| 6 | ∞ | ∞ | ∞ | 3 | ∞ | 0 |

Dalam graf berbobot, sering kali ingin mencari sebuah lintasan terpendek (yakni, sebuah lintasan yang mempunyai panjang minimum) antara dua verteks yang diketahui. Untuk graf berbobot tersambung, teknik pencarian lintasan terpendek diberikan oleh Algoritma Dijkstra. Algoritma Dijkstra melibatkan pemasangan label pada verteks. Misalkan L(v) menyatakan label dari verteks v. Pada setiap pembahasan, beberapa verteks mempunyai label sementara dan yang lain mempunyai label tetap.

Misalkan T menyatakan himpunan verteks yang mempunyai label sementara. Dalam menggambarkan algoritma tersebut, akan dilingkari verteks-verteks yang mempunyai label tetap. Selanjutnya akan di tunjukkan bahwa jika L(v) adalah label tetap dari verteks v, maka L(v) merupakan panjang lintasan terpendek dari a ke v. Sebelumnya semua verteks mempunyai label sementara. Setiap iterasi dari algoritma tersebut mengubah status satu label dari sementara ke tetap; sehingga dapat mengakhiri algoritma tersebut jika z menerima sebuah label tetap. Pada bagian ini L(z) merupakan panjang lintasan terpendek dari a ke z. Algoritma ini mencari

panjang lintasan terpendek dari verteks a ke z dalam sebuah graf berbobot tersambung. Bobot dari rusuk (i,j) adalah w(i,j)>0 dan label verteks x adalah L(x) (Johnsonbaugh, R., 1998). Hasilnya, L(z) merupakan panjang lintasan terpendek dari a ke z.

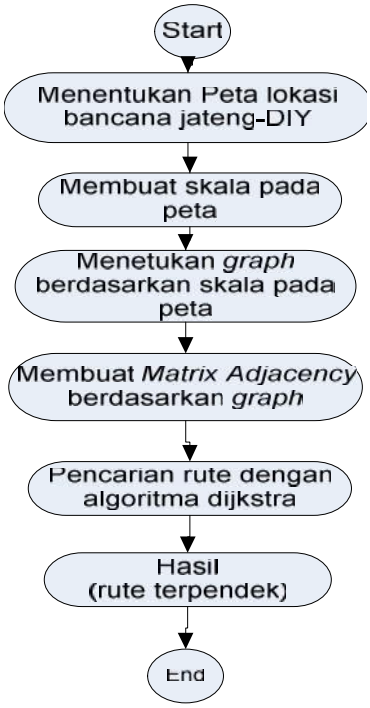
**Masukan**: Sebuah graf berbobot tersambung dengan bobot positif. Verteks a - z.

**Keluaran** : L(z), panjang lintasan terpendek dari a ke z.

1. **procedure** *dijkstra (w,a,z,L)*
2. L(a) := 0
3. **for** semua verteks x $\neq$a **do**
4. L(x) := $\infty$
5. T := himpunan semua verteks
6. // T adalah himpunan verteks yang panjang terpendeknya dari a belum ditemukan
7. **while** z$\in$T **do**
8. **begin**
9. pilih v$\in$T dengan minimum L(v)
10. T:=T-{v}
11. **for** setiap x $\in$T di samping v **do**
12. L(x):=min{L(x), L(v)+w(v,x)}
13. **end**
14. **end** *dijkstra*

*Node* dalam peta yang sudah membentuk *graph* berarah, dapat diketahui hubungan antar *node*. Hubungan antar *node* secara lengkap yang membentuk grap kemudian dibuat *matrix adjacency* yaitu hubungan antar *node* dan bobot.

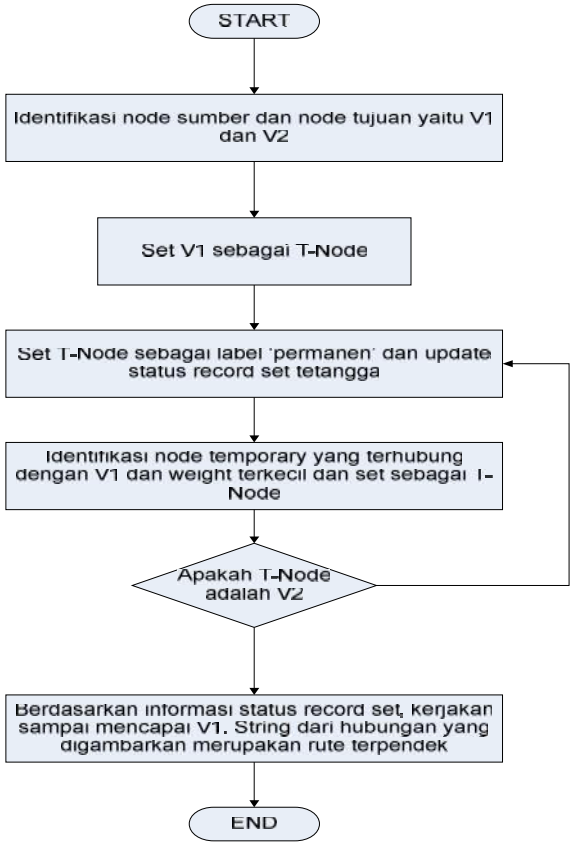
*Matrix adjacency* yang sudah terbentuk, kemudian dijalankan algoritma dijkstra. Dari algoritma dijkstra ini kemudian akan didapat hasilnya berupa rute terpendek. Sebagai gambaran terlihat pada Gambar 1.




**Gambar 1. *Flowchart* tahapan proses pencarian rute terpendek**

Ada beberapa aspek yang perlu diperhatikan pada proses pencarian rute. *Graph* yang sudah dibuat diidentifikasi *node* sumber dan *node* tujuan, notasi yang digunakan V1 dan V2. Kemudian dibuat *matrix adjacency* untuk koordinasi *weight*. Contohnya, [i,j] adalah *weight* yang terhubung antara Vi dan Vj. Jika tidak ada hubungan secara langsung antara Vi dan Vj, *weight* diidentifikasi *infinity*. Adapun langkah-langkahnya adalah:

1. Membuat status *record set* untuk setiap *node*. *Record set* berisi
   a. *Predecessor field*, yaitu *field* pertama menunjukan *node* sebelumnya.
   b. Panjang *field*, yaitu *field* kedua menunjukkan jumlah *weight* dari sumber ke *node*.
   c. Label *field* yaitu *field* terakhir menunjukkan status *node*. Setiap *node* dapat mempunyai status: ”Permanen” atau ”*temporary*”.
2. Parameter inisialisasi rute dari status *record set* (untuk semua *node*) dan set panjangnya ke “*infinity*” dan diberi label “*temporary*”.
3. Set rute sebagai T-*Node*. Contoh, Jika V1 adalah sumber T-*Node*, Rute merubah label V1 ke “permanent”, Ketika label berubah ke ”permanent”, itu tidak pernah berubah lagi. T-*Node* adalah perantara.
4. *Update* status *record set* rute untuk semua *node temporary* yang terhubung langsung ke T-*Node.*
5. Rute memperlihatkan semua *node temporary* dan memilihkan satu *weight* V1 terkecil. *Node* tersebut merupakan tujuan T-*Node*.
6. Jika *node* tidak V2 (tujuan yang diharapkan), rute kembali ke step 5. Jika *node* adalah V2, rute akan mencabut *node* sebelumnya dari status *record set* dan penyelesaikannya hingga sampai pada V1.



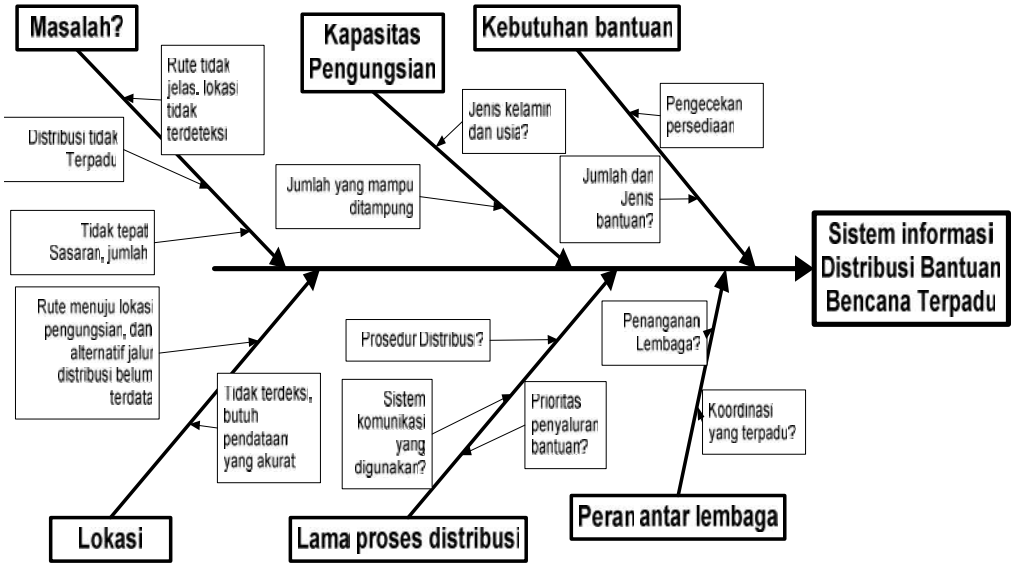

**Gambar 2. Langkah-langkah pencarian rute terpendek dengan Dijkstra**

Kelebihan dan kelemahan implementasi menggunakan algoritma Dijkstra, diantaranya adalah:

a. Algoritma Dijkstra dapat digunakan untuk memetakan jalur-jalur alternatif, apabila jalur utama mengalami hambatan.
b. Algoritma Djikstra dapat digunakan untuk menyelesaikan permasalahan rute terpendek dan aliran maksimum, elemen-elemen (bobot) dari rute tersebut berupa jarak tempuh, biaya, maupun hal lainnya.
c. Pada implementasi yang menggunakan algoritma Dijkstra sistem akan terputus dari *web server* jika terdapat suatu node pada *graph* yang berdiri sendiri atau tidak terhubung.

## METODOLOGI PENELITIAN

Kerangka berpikir sebab akibat penelitian mengenai sistem distribusi bantuan bencana dan rencana alur penelitian ditunjukkan pada Gambar 3. Manajemen bencana sebagai sebuah kepentingan bersama, dimana semua orang berharap berkurangnya korban nyawa dan kerugian harta benda. Hal yang terpenting dari manajemen bencana ini adalah adanya suatu langkah konkrit dalam mengendalikan bencana sehingga korban dapat terselamatkan dan upaya untuk pemulihan pasca bencana dapat dilakukan dengan cepat.

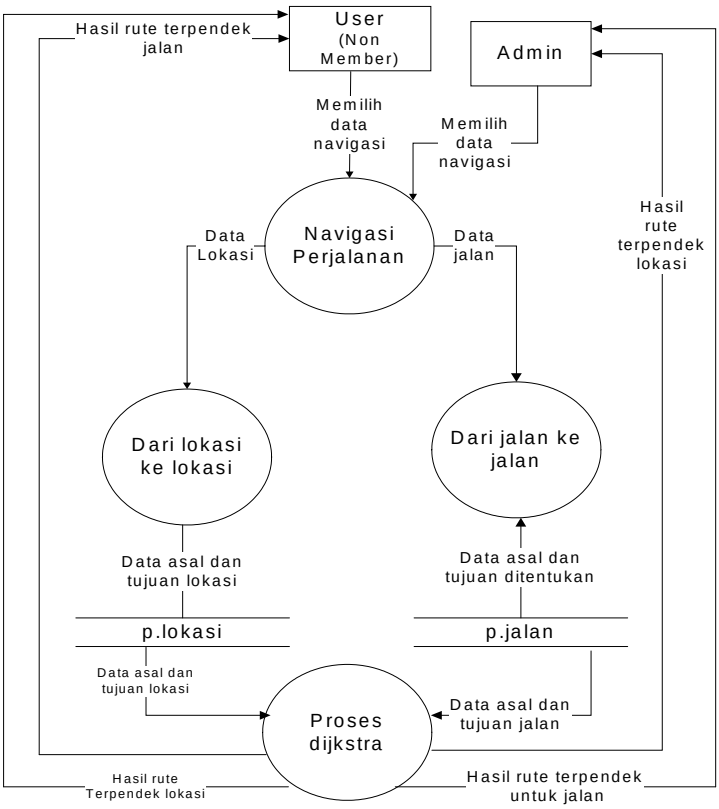



**Gambar 3. Diagram Sebab-Akibat Penelitian**

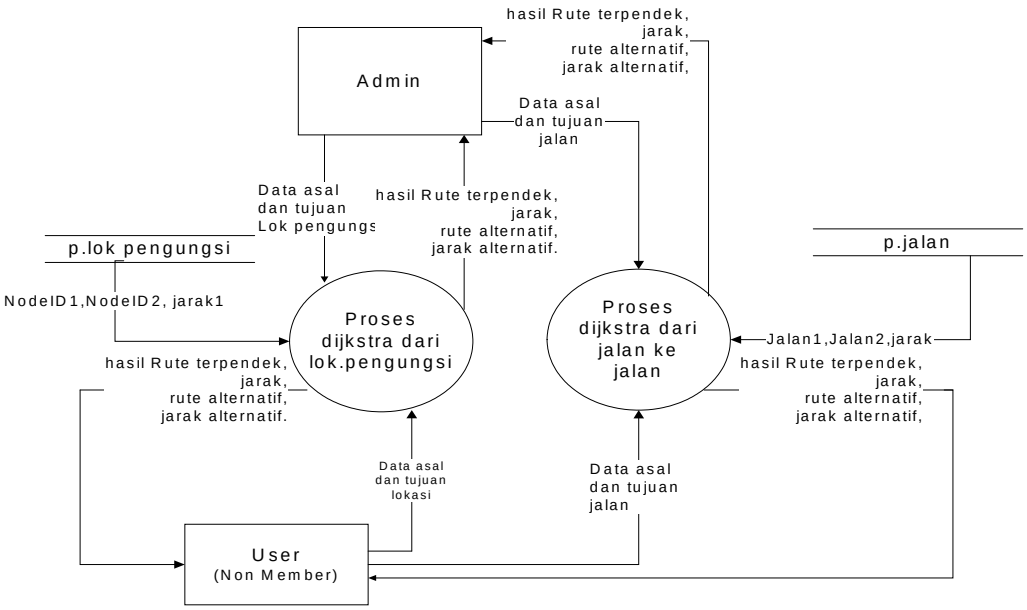
Pengendalian itu dimulai dengan membangun kesadaran kritis masyarakat dan pemerintah atas masalah bencana alam, menciptakan proses perbaikan atas pengelolaan bencana, penegasan untuk lahirnya kebijakan lokal yang bertumpu pada kearifan lokal yang berbentuk peraturan nagari dan peraturan daerah atas menejemen bencana. Yang tak kalah pentingnya dalam manajemen bencana ini adalah sosialisasi kehatian-hatian terutama pada daerah rawan bencana. Disamping itu pendistribusian bencana yang tidak merata akan mengakibatkan munculnya permasalah sosial yang baru, serta timbulnya konflik dan bahkan kematian karena kurangnya sumber makanan yang sangat dibutuhkan oleh para korban. Sehingga orientasi penelitian adalah:

1. Sistem perancangan distribusi bantuan yang lebih efektif, karena dilakukan koordinasi secara terpadu dengan lembaga-lembaga penanganan korban bencana yang terkait, sehingga informasi yang diberikan adalah informasi yang cepat, tepat, akurat dan terpercaya.
2. Kemampu sasaran yang lebih efektif dalam penyaluran bantuan bencana, sehingga tidak terjadi penumpukan dan kekurangan bantuan di masing-masing lokasi bencana.
3. Peningkatan kemampuan pendataan korban bencana beserta tempat pengungsian.
4. Penggunaan *handphone* mampu mengirimkan informasi secara cepat, dan sesuai dengan lokasi utama dan jalur alternatifnya, serta dilengkapi dengan berita kondisi rute yang akan dilalui.
5. Sistem juga menginformasikan data-data stok bantuan dan jenis kebutuhan dari para korban.
6. Meningkatnya motivasi masyarakat untuk membantu karena ada sistem yang menavigasi mereka menuju tempat-tempat pengungsian untuk menyalurka bantuan.

Realisasi rencana program dari penelitian adalah dengan menggunakan DFD. DFD merupakan diagram arus data (*data flow diagram*), atau DFD, adalah suatu gambaran grafis dari suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang berkaitan (McLeod, 2001). DFD merupakan cara paling alamiah untuk mendokumentasikan data dan proses. DFD untuk proses Djikstra adalah seperti terlihat pada Gambar 4 dan Gambar 5.

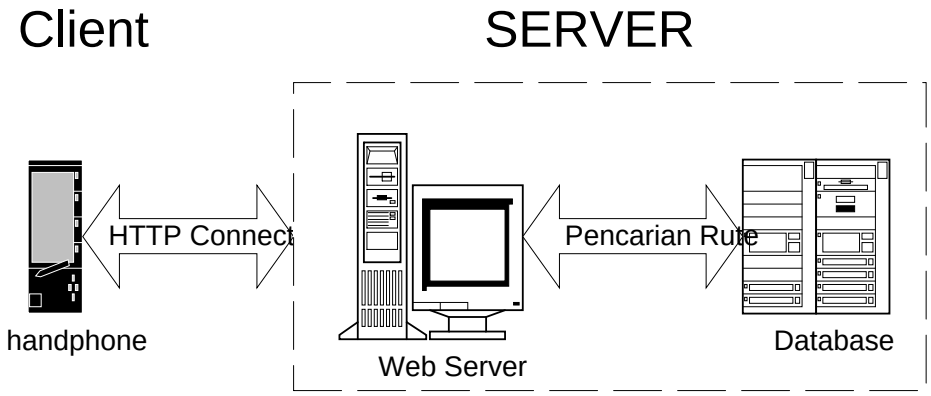



**Gambar 4. DFD Pada Proses pemilihan rute perjalanan**




**Gambar 5. DFD Pada Proses Djikstra**

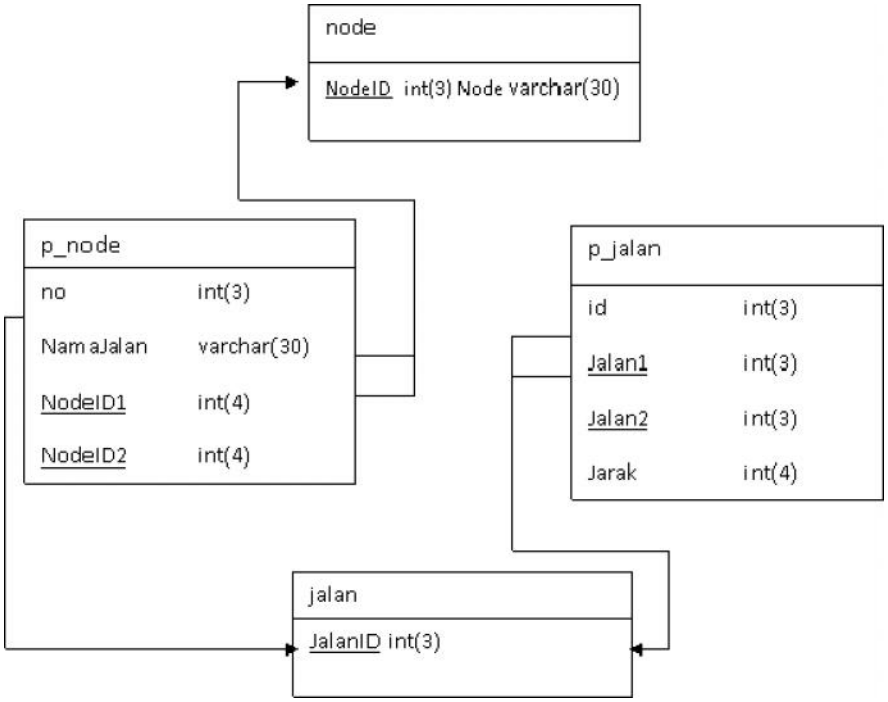
Sistem ini mempunyai dua sisi, yaitu sisi *client* dan sisi *server*. Sisi *client* merupakan user yang menggunakan *handphone* untuk navigasi perjalanan (pencarian rute). Sisi *server* terdiri dari web server yang menggunakan bahasa pemrograman PHP dan server database yang menggunakan mySQL. Ketika user memilih rute yang dikehendaki maka *handphone* akan mengirimkan *request* ke web server melalui *http connection*. Maka web server akan mencari rute yang tependek dengan mengambil data yang tersedia di database server dan hasilnya dikirim kembali ke *handphone*. Adapun gambaran dari arsitektur sistem navigasi pencarian rute terpendek ini terlihat pada Gambar 6.




**Gambar 6. Arsitektur sistem**

Spesifikasi minimal Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah, Prosesor *Intel Pentium* 4 CPU 2,4GHz, Memori 512MB, dan *hardisk* 30 GB. Perangkat lunak ini, data yang akan diolah, sebelumnya berada didalam *database*, *database* server yang digunakan adalah MySQL, data-data tersebut meliputi data-data atribut jalan. Data atribut yang akan diolah terdiri dari beberapa macam data, yaitu data jalan, dan data lokasi.

Dalam perancangan tabel lokasi dan jalan, data lokasi menggunakan node dan p_node. node berisi NodeID dan node, p_node terdiri no, NamaJalan, NodeID1, NodeID2 dan jarak, p_node di berikan atribut jarak karena untuk mendapatkan jarak harus diketahui dua buah node. Pada tabel jalan terdiri atas JalanID dan Jalan, Kemudian untuk p_jalan terdiri atas id, Jalan1, jalan2 dan jarak.




**Gambar 7. Perancangan relasi antar tabel Lokasi dan Jalan**

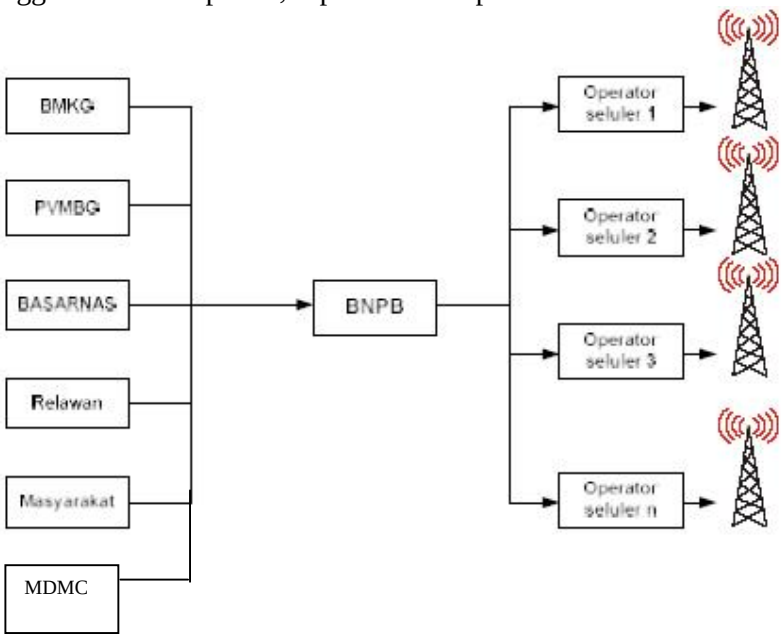
Untuk tabel user, terdiri dari user, Passwd_user, Nn_user, Jns_kelamin, No_hp, Tanggal_masuk.

| user | |
|---|---|
| User | varchar(25) |
| Passwd_user | varchar(25) |
| Nm_user | varchar (25) |
| Jns_kelamin | enum('P','W') |
| No_hp | varchar(15) |
| Tanggal_masuk | date |

**Gambar 8. Perancangan tabel *User***

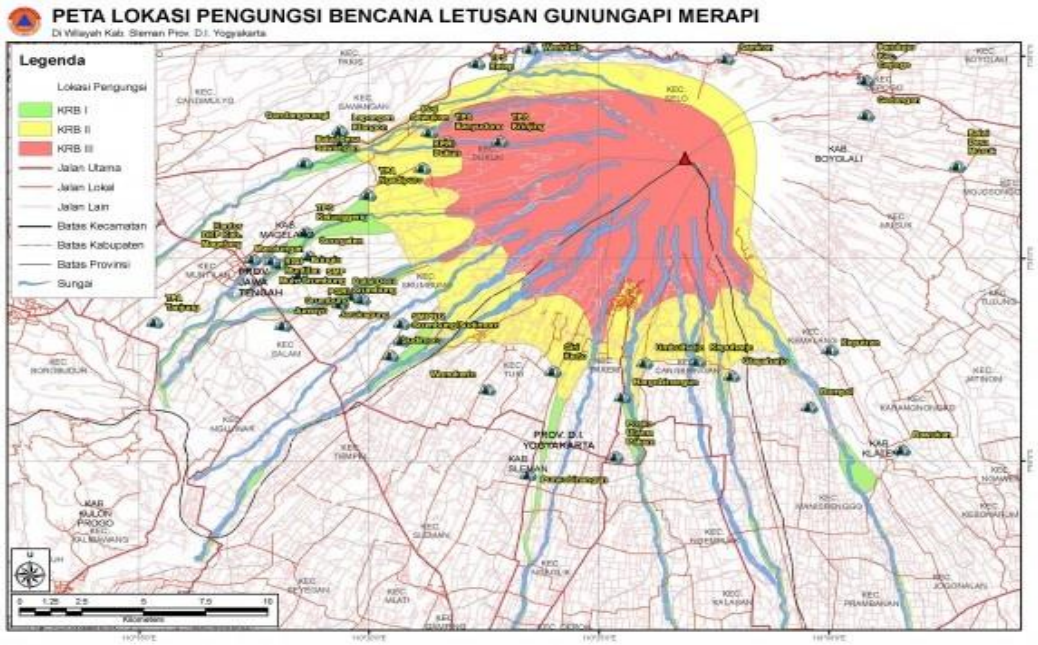
Skema mengenai arus informasi bencana diperlihatkan pada Gambar 6. Sumber informasi yang merupakan bagian input –BMKG, PVMBG, BASARNAS, MDMC, relawan dan masyarakat- yang memberikan informasi di lapangan kepada BNPB selaku badan yang dibentuk pemerintah untuk menangani masalah bencana. Untuk memastikan informasi BNPB melakukan cross check untuk menghindari kesalahan informasi dan selanjutnya informasi diteruskan ke sejumlah

operator seluler untuk disampaikan ke seluruh pelanggan di daerah yang dituju dengan menggunakan Handphone, seperti terlihat pada Gambar 8.




**Gambar 8. Blok diagram sistem penyebaran luasan informasi bencana alam**

Pengamatan awal pada beberapa lokasi pengungsian dan sumber yang didapatkan dari BNPB serta pemkab, dapat diketahui dari Gambar 9.




Gambar 9. Peta Lokasi Pengungsi Letusan Gunung Merapi di Wilayah Jateng-DIY

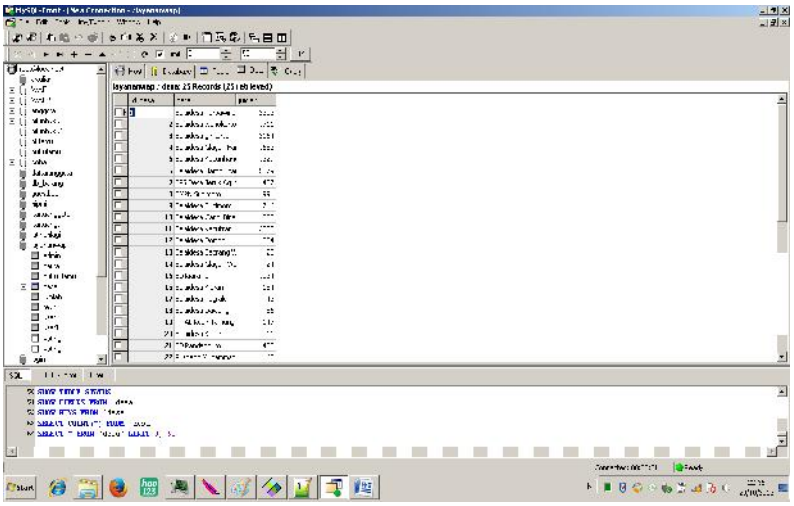
**Tabel 2. Daftar Tempat Pengungsian di Kabupaten Sleman**

| No | Tempat Pengungsian | No | Tempat Pengungsian |
|---|---|---|---|
| 1 | Balaidesa Banyurejo | 28 | Pondok Pesantren Al-Qodir |
| 2 | Balaidesa Margoagung | 29 | Balaidesa Candibinangun |
| 3 | Balaidesa Margokaton | 30 | Balaidesa Pakembinangun |
| 4 | Balaidesa Margomulyo | 31 | Balaidesa Harjobinangun |
| 5 | Balaidesa Sumberadi | 32 | Balaidesa Kiyaran |
| 6 | Balaidesa Tlogoadi | 33 | Balaidesa Argomulyo |
| 7 | Balaidesa Tirtoadi | 34 | Balaidesa Donoharjo |
| 8 | Youth Center | 35 | Balaidesa Umbulmartani |
| 9 | Balaidesa Merdikorejo | 36 | Kampus UII |
| 10 | Balaidesa Margorejo | 37 | PPG Kesenian |
| 11 | Balaidesa Mororejo | 38 | Balaidesa Sindumartani |
| 12 | Balaidesa Bangunkerto | 39 | Balaidesa Bimomartani |
| 13 | Balaidesa Trimulyo | 40 | Balaidesa Widodomartani |
| 14 | Balaidesa Caturharjo | 41 | Balaidesa Selomartani |
| 15 | Balaidesa Triharjo | 42 | Balaidesa Sinduharjo |
| 16 | Balaidesa Pendowoharjo | 43 | Balaidesa Minomartani |
| 17 | Gor Pangukan | 44 | Balaidesa Wedomartani |
| 18 | Masjid Agung Sleman | 45 | Balaidesa Taman martini |
| 19 | Balaidesa Sariharjo | 46 | Stadion Maguwoharjo |
| 20 | Balaidesa Sendangadi | 47 | Kampus UPN Veteran |
| 21 | Balaidesa Sardonoharjo | 48 | Kampus Sanata Darma |
| 22 | Asrama haji | 49 | Badan Diklat Keuangan |
| 23 | Kampus UGM | 50 | Bapelkes DIY |
| 24 | Balaidesa Donokerto | 51 | Balaidesa Bokoharjo |
| 25 | Balaidesa Wonokerto | 52 | LPMP Depdiknas |
| 26 | Balaidesa Girmbinangunikerto | 53 | Balaidesa Gayam |
| 27 | Balaidesa Purwobinangun | 54 | Balaidesa Glagaharjo |

Adapun kebutuhan yang paling dibutuhkan, namun tidak banyak sukarelawan maupun bantuan yang dapat menyediakan. Kebutuhan tersebut antara lain:

1. Minyak tanah
2. *Emergency lamp*
3. Makanan ringan untuk anak-anak
4. Bahan lauk pauk segar
5. Kasur lipat atau kasur lansia
6. Susu bayi
7. Makanan bayi
8. Handuk
9. Selimut
10. Pasta gigi
11. Sabun
12. Hiburan untuk anak-anak dan dewasa

Perencanaan database awal untuk sistem distribusi bantuan bencana diperlukan sebagai cara untuk mempermudah realisasi bantuan bencana, seperti terlihat pada Gambar 10.



**Gambar 10. Data barak pengungsian yang di masukkan dalam database**

Tampilan system informasi dalam aplikasi di handphone, terlihat pada Gambar 11.


**Gambar 11. Tampilan menu, data barak pengungsian, priopritas penerima bantuan dan jarak tempuh**

## KESIMPULAN

Setelah dilakukan serangkaian penelitian dan analisa terhadap perangkat lunak menggunakan algoritma djikstra yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Terciptanya sistem layanan informasi yang *real time*, dan sistem navigasi yang bisa diakses dengan telepon seluler yang mampu menunjukan rute yang paling pendek, serta dapat menunjukan rute alternatif jika terjadi kemacetan atau terjadi kecelakaan di salah satu ruas jalan atau di suatu lokasi bencana.
2. Algoritma Dijkstra dapat diimplementasikan untuk pendistribusian bantuan bencana mampu membuat informasi persediaan real time.
3. Hasil dari pencarian rute dengan memanfaatkan aplikasi WML akan mempermudah user dengan menunjukkan pedoman sesuai dengan hasil pencarian rute yang ada pada data.

## Daftar Pustaka

Johnsonbaugh, R. 1998. *Matematika Diskrit.* Jilid 2. Jakarta: Prenhallindo.

Librado, Dison. 2005. *Aplikasi WAP untuk Pendaftaran User Laboratorium dan Perpustakaan.* Tesis S2. Program Magister Ilmu Komputer, Universitas Gadjahmada Yogyakarta.

McLeod, R. Jr. 2001. *Management Information System*. Eight Edition. New Jersey: Prentice-Hall, Inc

Nandiroh, Siti. 2009. Penentuan Rute Terpendek Jalan dan Lokasi Pariwisata di Kota Surakarta Menggunakan Algoritma Dijkstra dan WAP Pada Handphone. *Jurnal Sains dan Teknologi*. ISSN 1411-5174, Vol.12, No.2 Oktober 2011. Universitas Muhammadiyah Surakarta.

Renaldi, M. 2001. *Matematika Diskrit.* (Buku Teks Ilmu Komputer). Bandung: Penerbit Informatika.