

## PERANCANGAN SISTEM PERSEDIAAN DAN DISTRIBUSI BANTUAN BENCANA YANG LEBIH EFEKTIF MENGGUNAKAN ALGORITMA DIJKSTRA

Siti Nandiroh<sup>1)</sup>, Haryanto<sup>2)</sup>, Hafidh Munawir<sup>3)</sup>

<sup>1,3)</sup> Jurusan Teknik Industri, Fakultas Teknik, Universitas Muhammadiyah Surakarta

<sup>2)</sup> Jurusan Sistem Komputer STMIK AUB Surakarta

Email: Siti.Nandiroh@ums.ac.id

### *Abstract*

*Research using communications technology in support Dijkstra algorithm, as a determinant of the distribution of aid is more effective for the volunteers and victims to determine the shortest route from street to street or from location to location, as well as alternate paths when there are obstacles in the path to the destination, and displays up to date news that may be filled by volunteers and staff as a member. The system is also equipped with a data inventory and needs in their respective camps, so there is no buildup and lack support in each of the refugee camps, as well as the proper distribution of aid, according to the type of victims' needs. The purpose of this research is to better facilitate and motivate the general public and access the presence helping disaster victims, as well as providing direct support for access to travel to the destination can be known in time quickly and accurately. Information distribution process starts from resources, such as the Meteorology and Geofisiksa (BMKG), the Center for Volcanology and Geological Hazard Mitigation (PVMBG), the National Search and Rescue Agency (Basarnas), Muhammadiyah Disaster Management Center (MDMC), the volunteers and the community. Further information is verified by the National Disaster Management Agency (BNPB).*

**Keywords :** *Dijkstra algorithm, relief, disaster, inventory systems*

### 1. 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 membutuhkan. Penyampaian informasi secara cepat dapat dilakukan jika menggunakan perangkat komunikasi yang canggih, perangkat komunikasi yang canggih saat ini diantaranya adalah telepon seluler.

### 2. KAJIAN LITERATUR

Algoritma *Dijkstra* diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini juga benar untuk graf tak berarah. (Munir, Renaldi, 2001). Algoritma yang di bahas dibawah ini adalah sebagai berikut:

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

$m_{ij}$  = bobot sisi  $(i,j)$  (pada graf berarah  $m_{ij}=m_{ji}$ )

$m_{ij} = 0$

$m_{ij} = \infty$ , 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  $i = 1, 2, \dots, n$

Langkah 1:

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

Langkah 2,3,...,n-1:

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

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

**Tabel 1. Matrix ketetanggaan**

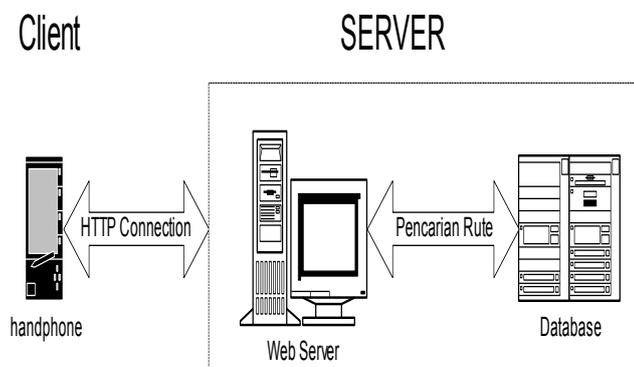
$j = 1$	2	3	4	5	6	
$i = 1$	0	50	10	40	45	$\infty$
2	$\infty$	0	15	$\infty$	10	$\infty$
3	20	$\infty$	0	15	$\infty$	$\infty$
4	$\infty$	20	$\infty$	0	35	$\infty$
5	$\infty$	$\infty$	$\infty$	30	0	$\infty$
6	$\infty$	$\infty$	$\infty$	3	$\infty$	0

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.

**Rancangan Sistem**

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 1 berikut ini :



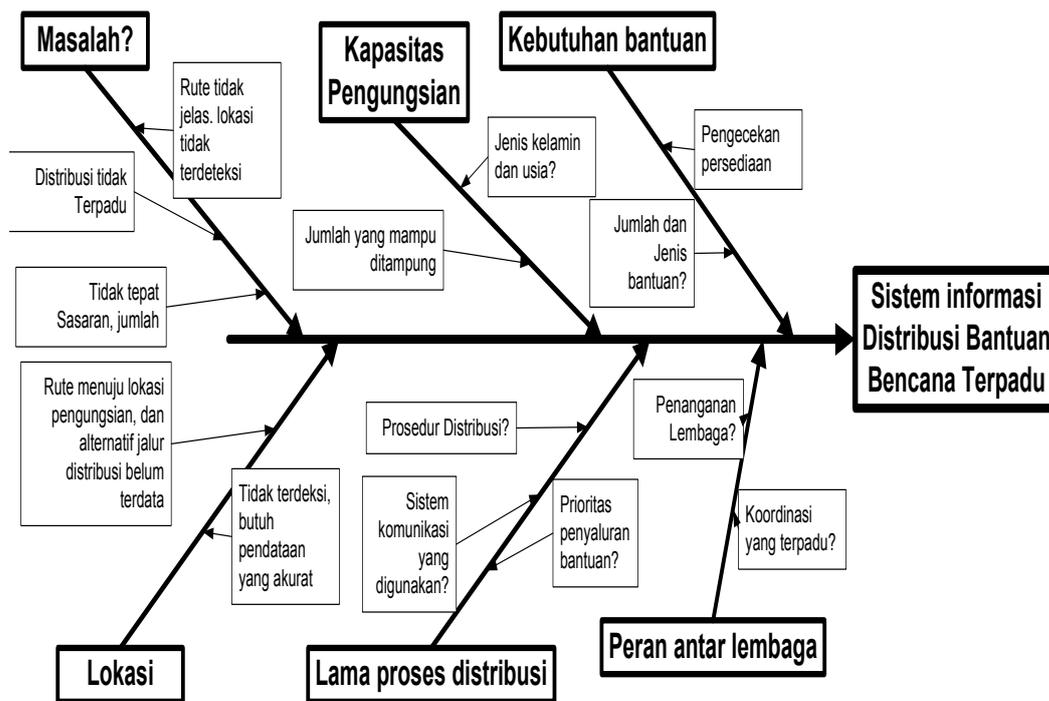
**Gambar 1. Arsitektur sistem**

Skema mengenai arus informasi bencana diperlihatkan pada Gambar 4.5. 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

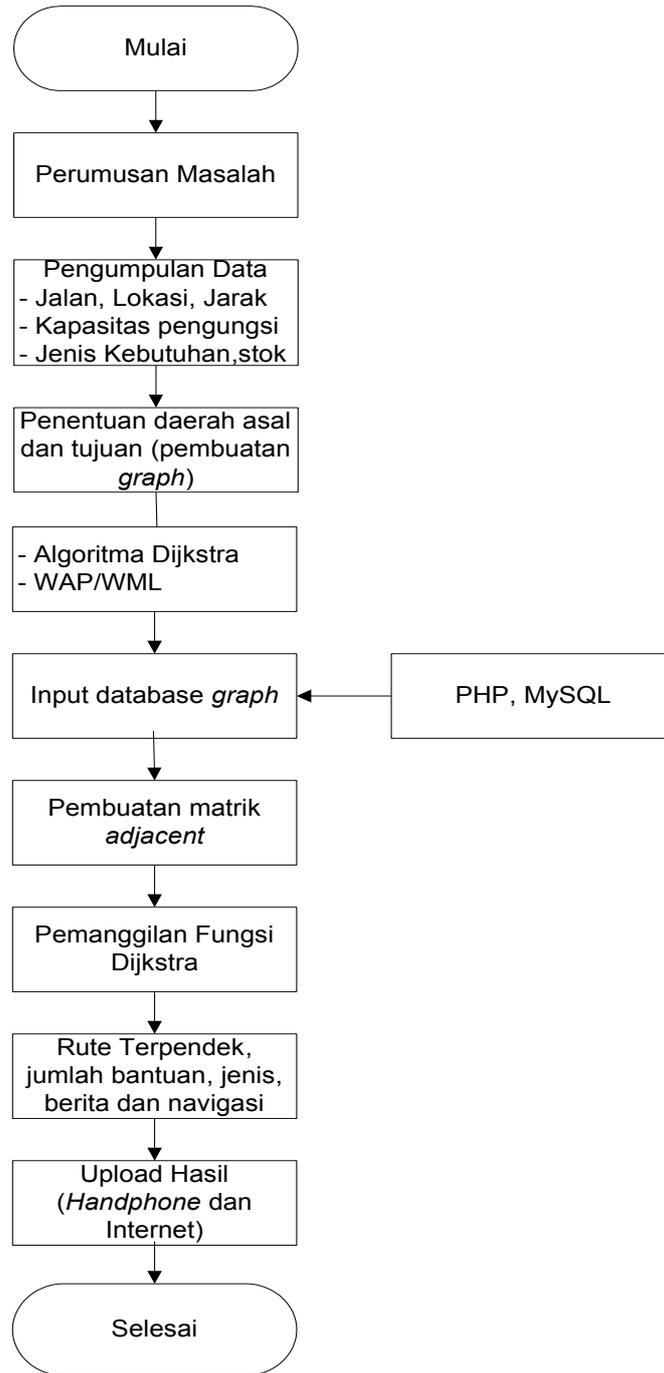
operator seluler untuk disampaikan ke seluruh pelanggan di daerah yang dituju dengan menggunakan Handphone.

### 3. METODOLOGI PENELITIAN

Kerangka berpikir penelitian mengenai sistem distribusi bantuan bencana dan rencana alur penelitian ditunjukkan pada Gambar 2, berikut ini,



Gambar 2.a. Fishbone Diagram Penelitian



Gambar 2.b. Alur Penelitian

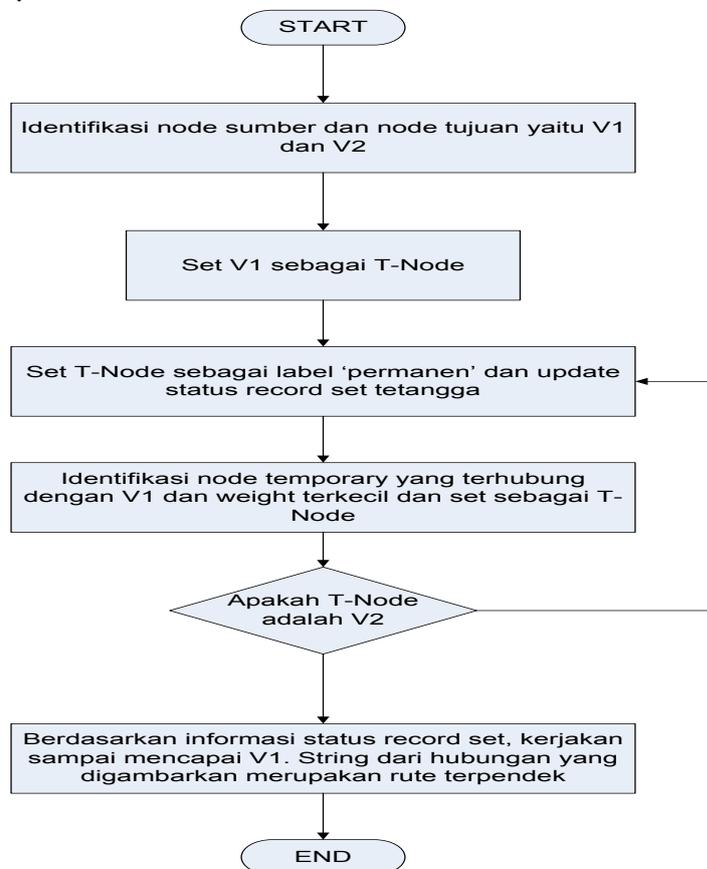
**Proses Pencarian Rute dengan Algoritma dijkstra**

Ada beberapa aspek yang perlu diperhatikan pada proses pencarian rute. *Graph* yang sudah dibuat diidentifikasi *node* sumber dan *node* tujuan, notasi yang

digunakan  $V_1$  dan  $V_2$ . Kemudian dibuat *matrix adjacency* untuk koordinasi *weight*. Contohnya,  $[i,j]$  adalah *weight* yang terhubung antara  $V_i$  dan  $V_j$ . Jika tidak ada hubungan secara langsung antara  $V_i$  dan  $V_j$ ,

*weight* diidentifikasi *infinity*. Adapun langkah-langkahnya adalah:

1. Membuat status *record set* untuk setiap *node*. *Record set* berisi
  - a. *Predecessor field*, yaitu *field* pertama menunjukkan *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 inialisasi 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.
7. Jika *node* adalah V2, rute akan mencabut *node* sebelumnya dari status *record set* dan menyelesaikannya hingga sampai pada V1.



Gambar 3.a Flowchart langkah-langkah Pencarian rute terpendek dengan dijkstra

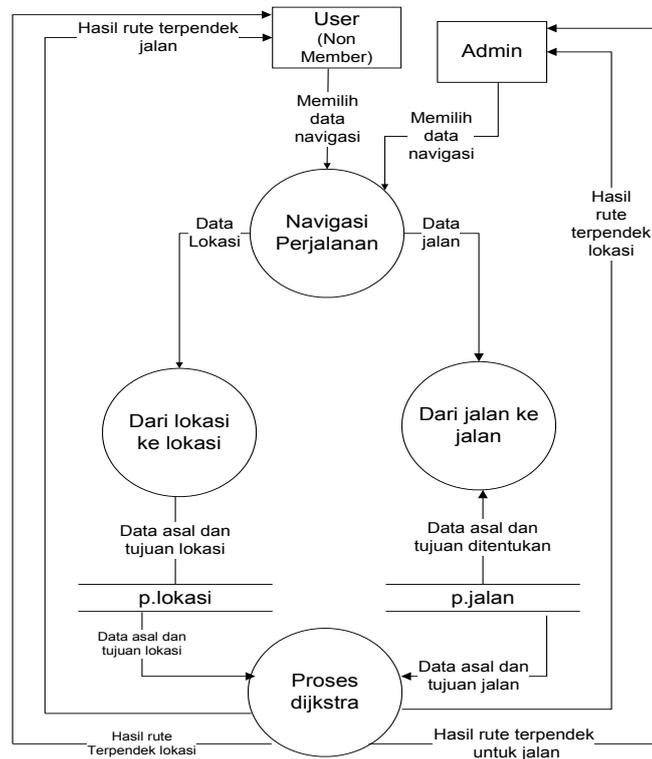
Kelebihan dan kelemahan implementasi menggunakan algoritma dijkstra , diantaranya adalah:

- a. Algoritma Dijkstra dapat digunakan untuk memetakan jalur-jalur alternatif, apabila jalur utama mengalami hambatan.
- b. Algoritma Dijkstra 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

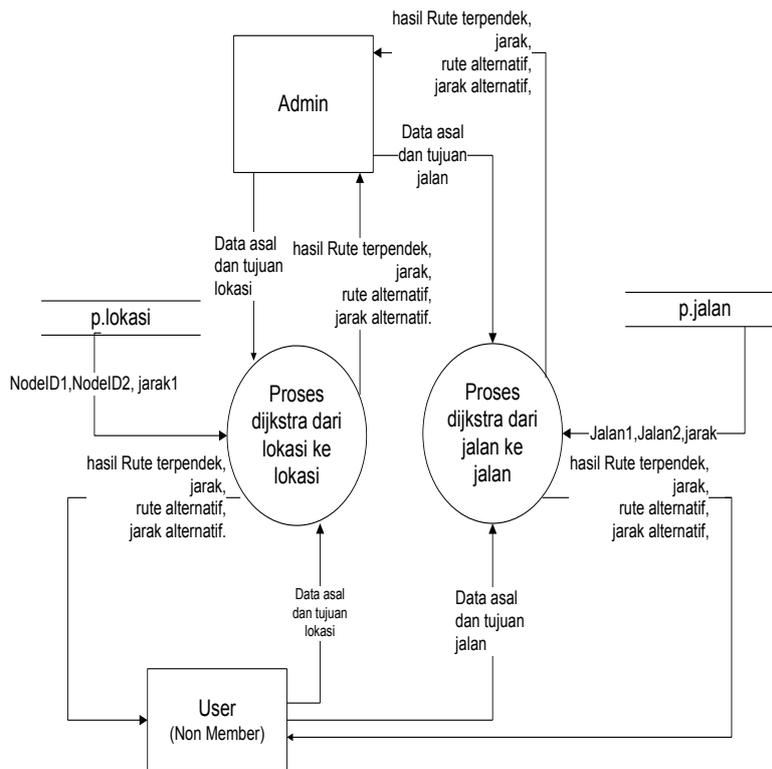
dari web server jika terdapat suatu node pada *graph* yang berdiri sendiri atau tidak terhubung.

**Data Flow Diagram**

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. Diagram konteks untuk sistem navigasi disajikan seperti Gambar berikut:



**Gambar 3.b DFD Level satu**

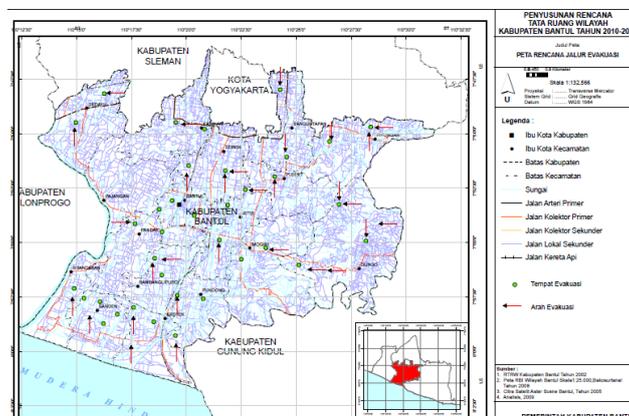


Gambar 3.c. DFD Pada Proses Dijkstra

**4. HASIL DAN PEMBAHASAN**  
**Tampilan Data Lokasi Pengungsian dan Kapasitas Pengungsi**

Pengamatan awal pada beberapa lokasi pengungsian dan sumber yang didapatkan

dari BNPB serta pemkab, dapat diketahui gambar-gambar peta lokasi berikut ini,

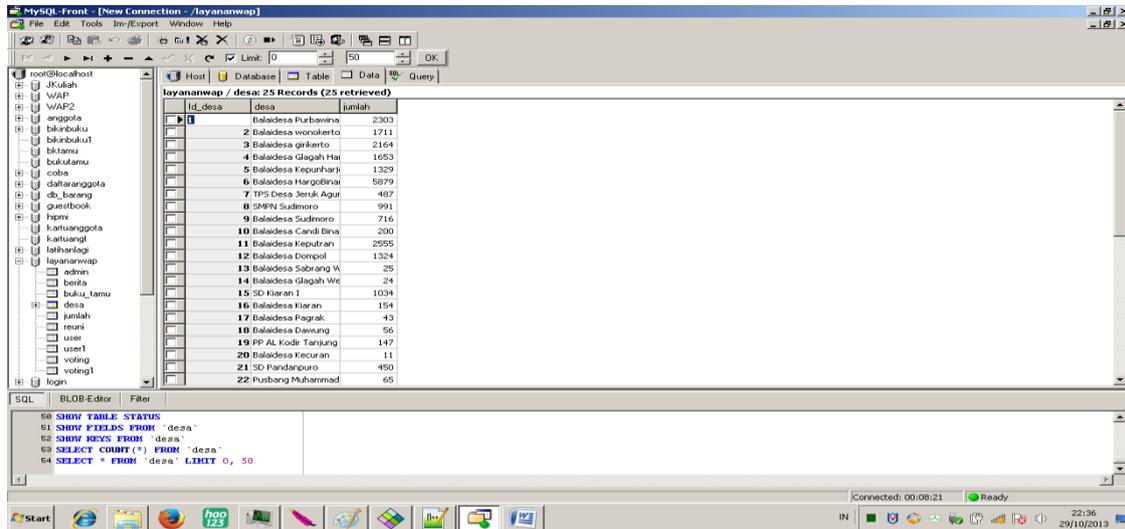


Gambar 4 .Gambar tampak penyebaran lokasi teridentifikasi

### Tampilan Sistem

Perencanaan database awal untuk sistem distribusi bantuan bencana diperlukan

sebagai cara untuk mempermudah realisasi bantuan bencana, seperti terlihat pada Gambar 5, berikut:



Gambar 5. Data barak pengungsian yang di masukkan dalam database

Data pada gambar 1, menunjukkan kapasitas pengungsi, wilayah pengungsian dan kode daerah, untuk melengkapi data status persediaan di masing-masing wilayah.

### Implementasi Halaman Utama

Halaman ini berisi menu yang menampilkan beberapa informasi seperti pencarian rute yang diinginkan di menu navigasi perjalanan, berita terbaru, info barak pengungsian, Prioritas Desa, Rute Desa dan status persediaan, seperti terlihat pada Gambar 6.

Halaman input adalah halaman yang menyediakan fasilitas agar *user member* atau *non member* dapat memasukkan data-data yang dibutuhkan untuk pencarian rute. Setelah memilih tipe pencarian rute, *user non member* harus mengisi data asal dan tujuan.

Asal adalah posisi awal dimana pencarian rute akan dilakukan, Asal dapat berupa jalan atau lokasi. Tujuan adalah posisi akhir dimana pencarian rute akan dilakukan, tujuan dapat berupa jalan atau lokasi. *user non member* bisa memilih menu dari lokasi ke lokasi atau dari jalan ke jalan.

Dalam *User member*, bisa melakukan input berita saja, *User member* dibatasi hanya user yang bertanggung jawab terhadap keamanan jalur dan lintasan. Kategori user semacam ini adalah kepolisian, DLLAJ, BNPB, Basarnas, MDMC, BMKG, dan Pemda terkait. *User member* berhak melakukan input berita, berita yang diinputkan secara *real time* akan bisa di lihat/diketahui informasinya bagi *user non member*, dengan cara melihat isi Berita Terbaru. Halaman input berita, terdapat dua *form* yaitu berita dan detail berita, kemudian setelah mengisi berita *user member* bisa



Gambar 6. Tampilan awal dari system

menyimpannya ke dalam database. Melalui tombol simpan. Setelah data berhasil disimpan maka selanjutnya isi berita ditampilkan ke dalam berita terbaru.

Dalam menu input berita, *user member* hanya bisa memasukkan data. Untuk Melihat Berita Terbaru *user member* harus *logout* terlebih dahulu, setelah *logout user member* bisa mengakses seperti *user non member*.

Untuk masuk pada halaman Administrasi, maka seorang admin harus mempunyai autentifikasi berupa Login admin dan passwordnya. Input teks administrator Pada halaman [Input Berita], [Berita Terbaru], [Navigasi Perjalanan], sama seperti pada menu *user* dan *user non member*. Selanjutnya untuk halaman [Input P\_node] merupakan penambahan data untuk lokasi.

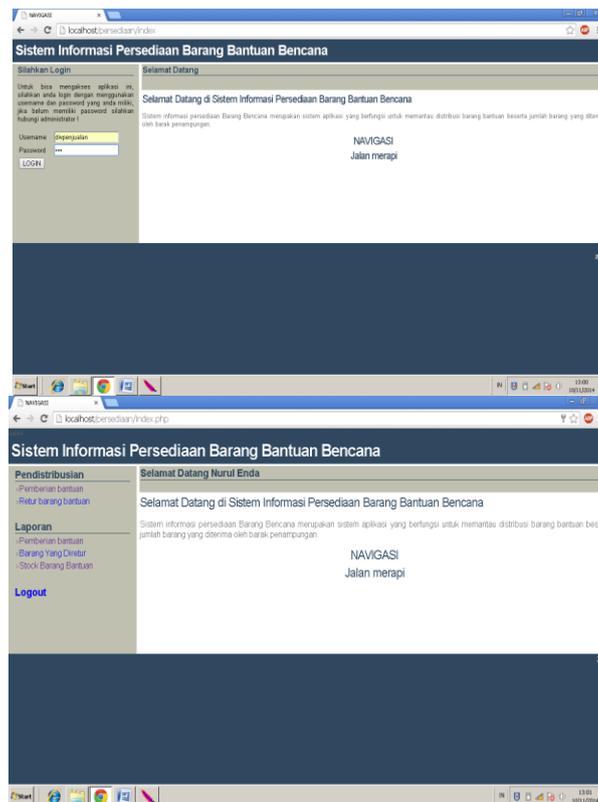
Adapun tampilan dalam sistem informasi berbasis handphone yang berhasil dibuat seperti terlihat pada Gambar 7, berikut ini.



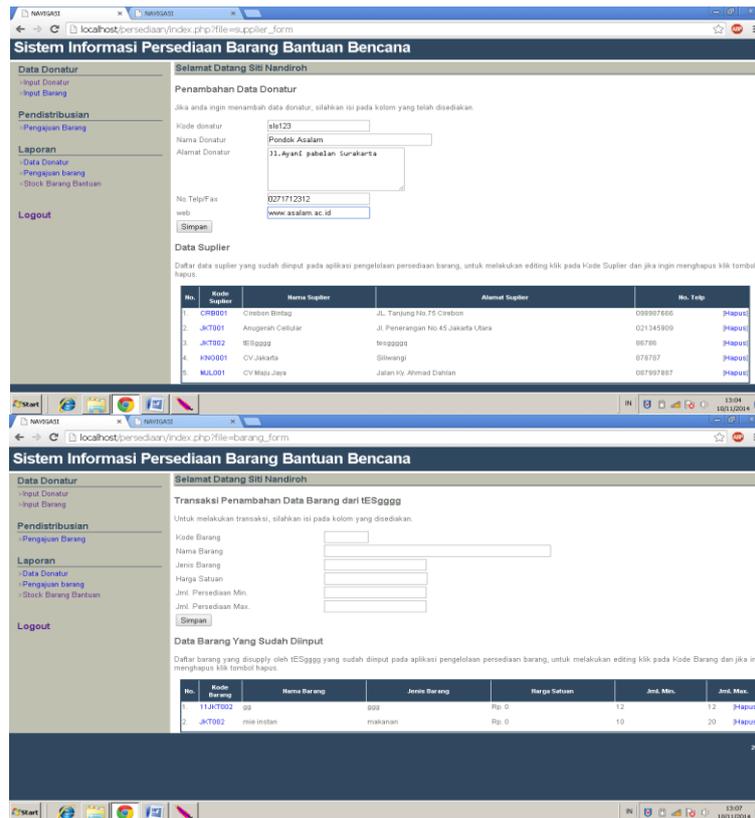
Gambar 7. Prioritas penerima bantuan

**Tampilan Database Sistem Persediaan**

Persediaan untuk masing-masing daerah pengungsian, tergantung dari integrasi informasi antara semua pihak. Regulasi dan kapasitas menjadi sumber data sistem informasi, seperti terlihat pada Gambar 8 dan 9. berikut ini,



Gambar 8. halaman utama sistem persediaan



Gambar 9. Menu Data base Persediaan

## 5. SIMPULAN

Perancangan sistem pada penelitian ini mempunyai hasil sebagai berikut:

1. Membuat sistem persediaan bantuan yang up to date.
2. Integrasi Sistem navigasi tanggap bencana terpadu yang bisa di akses dengan telepon seluler bagi pengguna seluler
3. Membuat informasi berita terbaru, jika terjadi sesuatu yang tidak diinginkan, maka user akan melewati jalan alternatifnya.
4. Member dapat Melakukan *up date* data persediaan dan jenis kebutuhan yang diinginkan para korban bencana, di wilayah pengungsian, sehingga dapat diketahui kelebihan dan kekurangan stok untuk masing-masing tempat pengungsian.

## PERSANTUNAN

Banyak pihak yang memberikan semangat, bimbingan dan arahan, sehingga hambatan dan kesulitan dalam penyelesaian penelitian ini dapat teratasi. Pada kesempatan

ini penulis mengucapkan terimakasih yang sebesar-besarnya kepada yang terhormat:

1. DP2M DIKTI, Selaku pemberi dana
2. Prof. Dr. Bambang Setiaji, selaku rektor UMS
3. Dr. Agus Ulinuha, selaku ketua LPPM UMS.
4. Ir Sri Sunarjono, MT, PhD selaku Dekan Fakultas Teknik UMS
5. Hafidh Munawir ST.MEng selaku Ketua Jurusan Teknik Industri.
6. Civitas Akademika di Jurusan Teknik Industri UMS

## REFERENSI

- Arunanto, F.X; Ulum, A dan Johan, A .F., (2006), *Perancangan dan Pembuatan Perangkat Lunak Pencarian Rute Tercepat menggunakan SMS*, Prosiding Seminar nasional Pascasarjana VI FTIF ITS, Surabaya, ISBN-979-545-027-1.
- Djojo, Minarto., (2000), *Perkembangan Internet Pada Mobile Device, Arcl*

- Technologies*, <http://www.arcle.net>.  
diakses 1-9-2007.
- Librado, Dison., (2005), *Aplikasi WAP untuk pendaftaran user Laboratorium dan Perpustakaan*, Tesis S2 Program Magister Ilmu Komputer, Universitas Gadjahmada Yogyakarta.
- Nandiroh, Siti., (2009), *Penentuan Rute Terpendek Jalan dan Lokasi Pariwisata di Kota Surakarta Menggunakan Algoritma dijkstra dan WAP Pada Handphone*, Jurnal Sains dan Teknologi UMS