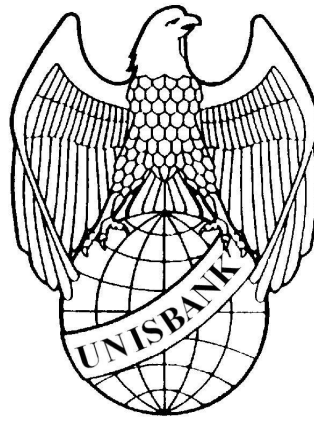


APLIKASI INFORMASI CUACA DAN GEMPA BUMI PADA BMKG SEMARANG BERBASIS SMS GATEWAY

Tugas Akhir disusun untuk memenuhi syarat
Mencapai gelar Kesarjanaan Komputer pada
Program Studi Teknik Informatika
Jenjang Program Strata-1



oleh:

PURWANTO
08.01.53.0016

10247

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK)**

SEMARANG

2013

PERNYATAAN KESIAPAN UJIAN SKRIPSI

Saya, Purwanto, dengan ini menyatakan bahwa Laporan Skripsi yang berjudul:

Aplikasi Informasi Cuaca dan Gempa Bumi Pada BMKG Semarang Berbasis SMS Gateway

Adalah benar hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain.

(Purwanto)
NIM : 08.01.53.0016

Disetujui oleh Pembimbing
Kami setuju Laporan tersebut diajukan untuk Ujian Skripsi

Semarang : 18 Pebruari 2013

(Dewi Handayani UN, S.Kom, M.Kom)
Pembimbing

HALAMAN PENGESAHAN

Telah dipertahankan di depan tim dosen penguji Tugas Akhir Fakultas Teknologi Informasi, Universitas STIKUBANK (UNISBANK) Semarang dan diterima sebagai salah satu syarat guna menyelesaikan jenjang Program Strata 1, Program Studi Teknik Informatika.

Semarang : 18 Maret 2013

Ketua

Sunardi, S.Kom, M.Cs

Sekretaris

Imam Husni Al Amin, S.T, M.Kom

Anggota

Agung Prihandono, S.Kom

MENGETAHUI :
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG

Fakultas Teknologi Informasi
Dekan

Dwi Agus Diartono, S.Kom, M.Kom

MOTTO DAN PERSEMBAHAN

MOTTO

- ✍ Hidup adalah perjuangan
- ✍ Sabar menghadapi masalah dan bersyukur merupakan salah satu pedoman hidup manusia
- ✍ Orang yang bahagia adalah orang yang dijauhkan dari fitnah dan bila dtimpa ujian serta cobaan ia selalu bersabar
- ✍ Lupakan kesempurnaan, dan cobalah mengejar kesempurnaan
- ✍ Sesungguhnya setelah mengalami kesulitan selalu ada jalan kemudahan
- ✍ Tiada hari tanpa instropeksi diri
- ✍ buku merupakan jendela informasi dunia

PERSEMBAHAN

1. Allah S.W.T
2. Teman-teman Teknik Informatika.
3. Sahabat–sahabat dan saudara –saudara yang memberi semangat.

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK) SEMARANG**

Program Studi : Teknik Informatika
Tugas Akhir Sarjana Komputer
Semester Ganjil Tahun 2013

**Aplikasi Informasi Cuaca dan Gempa Bumi
Pada BMKG Semarang Berbasis SMS Gateway**

**Purwanto
NIM : 08.01.53.0016**

Abstrak

BMKG atau sering disebut dengan Badan Meteorologi, Klimatologi dan Geofisika merupakan sebuah instansi yang menangani tentang masalah iklim, tsunami, informasi cuaca di kota tertentu, informasi posisi gempa bumi terbaru, suhu udara, kelembaban, lama penyinaran matahari, tekanan udara dan lain sebagainya dalam kurun waktu tertentu maupun data sekunder seperti prakiraan cuaca esok hari, prakiraan cuaca seminggu kedepan, iklim dan lain-lain yang dapat digunakan untuk tujuan penelitian.

Adapun tujuan yang akan dicapai adalah membuat aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dengan menggunakan Delphi dan XML sehingga masyarakat dapat mengetahui informasi tentang cuaca dan gempa bumi hanya dengan melalui SMS.

Hasil dalam penelitian ini adalah aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dibuat dengan menggunakan Delphi dan XML dengan menggunakan engine GAMMU untuk menghubungkan modem dengan database MySQL.

Kata Kunci

BMKG, Cuaca dan Gempa Bumi, SMS Gateway

Semarang : 12 Pebruari 2013

Pembimbing

(Dewi Handayani UN, S.Kom, M.Kom)

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT. yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya kepada penulis sehingga laporan tugas akhir dengan judul **“Aplikasi Informasi Cuaca dan Gempa Bumi Pada BMKG Semarang Berbasis SMS Gateway”** dapat penulis selesaikan sesuai dengan rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terimakasih kepada:

1. Dr. Bambang Suko Priyono, MM selaku Rektor Universitas Stikubank Semarang.
2. Dwi Agus Diartono, S.Kom, M.Kom, selaku Dekan Fakultas Teknologi Informasi.
3. Dewi Handayani UN, S.Kom, M.Kom selaku Ka. Progdik Teknik Informatika.
4. Dewi Handayani UN, S.Kom, M.Kom selaku Pembimbing yang telah membantu dan memberikan bimbingan serta pengarahan dalam penulisan Laporan Tugas Akhir ini.
5. Dosen-dosen pengampu di Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Stikubank Semarang yang telah memberikan ilmu dan pengalamannya masing-masing, sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.

Semoga Allah SWT memberikan balasan yang lebih besar kepada beliau-beliau, dan pada akhirnya penulis berharap bahwa penulisan laporan tugas akhir ini dapat bermanfaat dan berguna sebagaimana fungsinya.

Semarang, Pebruari 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
MOTTO DAN PERSEMBAHAN	iv
ABSTRAKSI	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
 BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	3
1.4 Metodologi Penelitian	4
1.4.1. Objek Penelitian	4
1.4.2. Jenis Data	4
1.4.3. Metode Pengumpulan Data	5
1.4.4. Metode Pengembangan Sistem	5
1.5 Sistematika Penulisan	7
 BAB II TINJAUAN PUSTAKA	
2.1 Pustaka Yang Terkait Dengan Penelitian.....	10
2.2 Perbedaan Penelitian Yang Dilakukan Dengan Penelitian Terdahulu	14
 BAB III LANDASAN TEORI	
3.1 Cuaca.....	15
3.2 Gempa Bumi	17

3.3.	SMS Gateway	18
3.3.1.	SMS (Short Message Service)	18
3.3.2.	SMS Gateway	18
3.3.3.	Komponen Pendukung SMS Gateway.....	19
3.3.4.	Keuntungan SMS Gateway	20
3.3.5.	Kekurangan SMS Gateway	20
3.3.6.	Model SMS Gateway	21
3.4.	Desain Berorientasi Objek	21
3.4.1.	Use Case.....	22
3.4.2.	Activity Diagram.....	23
3.5.	Delphi.....	24
3.5.1.	Integrated Development Environment	24
3.5.2.	Menu Bar.....	25
3.5.3.	Tool Bar	27
3.5.4.	Component Palette	28
3.5.5.	Object Inspector	28
3.5.6.	Form	29
3.5.7.	Code Editor	30
3.5.8.	Tipe Data Pada Delphi	30
3.5.9.	Konversi Tipe Data	31
3.6.	Gammu.....	31
3.7.	XML.....	34
3.8.	MySQL.....	36
3.8.1.	Keistimewaan MySQL.....	36

BAB IV ANALISA DAN PERANCANGAN SISTEM

4.1	Gambaran Umum Perusahaan.....	39
4.1.1.	Sejarah Perusahaan.....	39
4.1.2.	Visi, Misi dan Tupoksi.....	41
4.1.3.	Struktur Organisasi	44
4.2	Analisa Sistem.....	44

4.2.1.	Analisa Permasalahan	44
4.3	Perancangan Sistem	45
4.3.1.	Deskripsi Sistem	45
4.3.2.	Use Case Diagram.....	47
4.3.3.	Activity Diagram.....	48
4.3.4.	Arsitektur Sistem.....	50
4.3.5.	Desain Input Output	51
4.3.6.	Format SMS	53
BAB V	IMPLEMENTASI SISTEM	
5.1	Menu Utama	54
5.2	Menu Inbox	60
5.3	Menu Outbox	61
5.4	Menu PhoneBook.....	62
5.5	Kebutuhan Sistem	63
5.5.1.	Kebutuhan Perangkat Keras.....	63
5.5.2.	Kebutuhan Perangkat Lunak.....	63
BAB VI	HASIL PENELITIAN DAN PEMBAHASAN	
6.1	Hasil Penelitian	65
6.2.	Pembahasan.....	71
BAB VII	PENUTUP	
7.1	Kesimpulan	74
7.2.	Saran.....	75

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 3.1.	Simbol Use Case	22
Tabel 3.2.	Simbol Activity Diagram	24

DAFTAR GAMBAR

Gambar 3.1.	Alur SMS Gateway	20
Gambar 3.2.	Tampilan IDE Delphi.....	25
Gambar 3.3.	Menu Bar.....	26
Gambar 3.4.	Tool Bar	28
Gambar 3.5.	Component Palette	28
Gambar 3.6.	Object Inspector	29
Gambar 3.7.	Form	29
Gambar 3.8.	Code Editor	30
Gambar 3.9.	Gammu.....	32
Gambar 4.1.	Struktur Organisasi	44
Gambar 4.2.	Use Case Diagram.....	48
Gambar 4.3.	Activity Diagram.....	49
Gambar 4.4.	Arsitektur Sistem.....	50
Gambar 4.5.	Menu Utama.....	51
Gambar 4.6.	Menu Inbox	52
Gambar 4.7.	Menu Outbox	52
Gambar 4.8.	Menu PhoneBook.....	53
Gambar 5.1.	Menu Utama.....	54
Gambar 5.2.	Menu Inbox	60
Gambar 5.3.	Menu Outbox	61
Gambar 5.4.	Menu PhoneBook.....	62
Gambar 6.1.	Informasi Cuaca di Website BMKG.....	66
Gambar 6.2.	Inbox Permintaan Informasi Cuaca	66
Gambar 6.3.	Outbox Balasan Cuaca	67
Gambar 6.4.	SMS Balasan Cuaca	67
Gambar 6.5.	Informasi Gempa Pada Website BMKG.....	68
Gambar 6.6.	Inbox Permintaan Informasi Gempa	69
Gambar 6.7.	Outbox Balasan Gempa.....	69
Gambar 6.8.	SMS Balasan Gempa	70

Gambar 6.9.	SMS Format Salah	70
Gambar 6.10.	Informasi Cuaca Pada Website BMKG	71
Gambar 6.11.	SMS Cuaca dan XML Cuaca BMKG	72
Gambar 6.12.	Informasi Gempa Pada website BMKG.....	72
Gambar 6.13.	SMS Gempa dan XML Gempa BMKG	73

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi informasi dan komunikasi telah mempengaruhi peradaban yang memungkinkan pekerjaan-pekerjaan di dalam suatu organisasi dapat diselesaikan secara cepat, akurat dan efisien. Salah satu bidang teknologi yang berkembang sangat cepat adalah teknologi komunikasi yang meliputi perangkat keras seperti komputer, alat-alat elektronik semakin banyak digemari oleh pengguna, dikarenakan dapat memenuhi kebutuhan akan sesuatu yang nyaman dan efisien. Telepon seluler (ponsel) mempunyai kelebihan yang bisa dibawa kemana-mana baik di kantor, di rumah, di kampus, di jalan atau di tempat lainnya, sehingga seseorang dapat saling berkomunikasi dengan cepat tanpa dibatasi ruang atau posisi dimana seseorang itu berada. Tentunya dengan catatan selama di dalam area operator ponsel itu sendiri. Sehingga tak diragukan lagi, ponsel memang sangat penting sekali keberadaannya.

Salah satu indikasi tersebut adalah munculnya layanan seperti pesan data pendek atau *Short Message Service* (SMS) pada sistem GSM. Orang tidak pernah menyangka layanan SMS sedemikian tinggi dan disukai orang. Kegemaran pengguna ponsel dalam ber-SMS yang lebih murah, praktis dan

terdapat pada semua jenis serta tipe ponsel, membuat fitur yang satu ini tetap digemari dan bertahan hingga saat ini.

BMKG atau sering disebut dengan Badan Meteorologi, Klimatologi dan Geofisika merupakan sebuah instansi yang menangani tentang masalah iklim, tsunami, informasi cuaca di kota tertentu, informasi posisi gempa bumi terbaru, suhu udara, kelembaban, lama penyinaran matahari, tekanan udara dan lain sebagainya dalam kurun waktu tertentu maupun data sekunder seperti prakiraan cuaca esok hari, prakiraan cuaca seminggu kedepan, iklim dan lain-lain yang dapat digunakan untuk tujuan penelitian.

Perancangan aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* merupakan pemecahan dari masalah tersebut yang dapat mempermudah dan mempercepat pihak BMKG dalam menginformasikan tentang cuaca dan bencana alam yang terjadi kepada masyarakat sehingga dapat mengetahui informasi tentang cuaca dan gempa bumi. Oleh karena itu dibuatlah **“Aplikasi Informasi Cuaca dan Gempa Bumi Pada BMKG Semarang Berbasis SMS Gateway”**.

1.2. Perumusan Masalah

Berdasarkan latar belakang permasalahan diatas, penulis dapat membuat suatu rumusan masalah yaitu bagaimana merancang aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dengan menggunakan Delphi dan XML sehingga masyarakat dapat

mengetahui informasi tentang cuaca dan gempa bumi hanya dengan melalui SMS. Adapun pembatasan masalah pada penelitian ini yaitu :

1. Aplikasi ini hanya menampilkan informasi cuaca dan gempa bumi yang bersumber dari website www.bmkgjateng.com.
2. Format SMS yang dikirimkan sesuai dengan format yang dibuat oleh BMKG.
3. Aplikasi ini dibuat dengan menggunakan Delphi dan XML.
4. Engine SMS menggunakan Gammu versi 13.

1.3. Tujuan dan Manfaat Penelitian

Tujuan penelitian yang ingin dicapai adalah membuat aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dengan menggunakan Delphi dan XML sehingga masyarakat dapat mengetahui informasi tentang cuaca dan gempa bumi hanya dengan melalui SMS. Hasil dari penelitian ini diharapkan nantinya dapat bermanfaat bagi banyak pihak, diantaranya :

1. Bagi BMKG Semarang

Hasil dari penelitian ini diharapkan akan dijadikan sebagai informasi dan merupakan sumbangan yang berarti bagi pihak BMKG Semarang untuk sekarang dan kedepannya khususnya mengenai informasi cuaca dan gempa bumi sehingga dapat mempermudah masyarakat untuk mengetahui informasi tentang cuaca dan gempa bumi.

2. Bagi Akademik

Dapat dijadikan sebagai bahan referensi bagi mereka yang mengadakan penelitian untuk dikembangkan lebih lanjut dengan masalah yang berbeda.

3. Bagi Penulis

Dapat menerapkan dan mengembangkan ilmu yang diperoleh di bangku kuliah melalui kasus nyata serta menambah wawasan sehingga memungkinkan mempertinggi kemampuan serta penguasaan dari penelitian yang dibuat.

1.4. Metodologi Penelitian

Metodologi penelitian merupakan suatu cara atau prosedur yang digunakan untuk mengumpulkan data, dengan perantara teknik tertentu. Dalam penulisan skripsi ini, akan menggunakan beberapa metode penelitian yaitu :

1.4.1. Objek Penelitian

Penelitian ini dilakukan di BMKG Semarang yang beralamat di jalan Siliwangi no 291 Semarang.

1.4.2. Jenis Data

Dari metode pengumpulan data tersebut didapatkan data yang diklasifikasikan sebagai berikut:

1. Data Primer

Data primer adalah data yang diperoleh melalui wawancara maupun pengamatan secara langsung dari sumber data yang bersangkutan misalnya data mengenai sejarah, struktur organisasi, tugas pokok dan lain sebagainya didapat secara langsung melalui penelitian di BMKG Semarang.

2. Data Sekunder

Data sekunder yaitu data yang diperoleh secara tidak langsung atau melalui studi literature dan dokumen yang berkaitan dengan penelitian seperti artikel-artikel dan dokumen tentang penelitian yang dibuat dan dokumen tentang pemograman Delphi dan XML.

1.4.3. Metode Pengumpulan Data

Metode yang digunakan dalam proses pengumpulan data sebagai bahan pembuatan sistem adalah:

1. Wawancara

Metode yang dilakukan dengan cara mengadakan wawancara langsung dengan pihak yang bersangkutan. Kegiatan yang dilakukan adalah melakukan wawancara dengan bagian data dan informasi pada BMKG Semarang mengenai proses penyampaian informasi cuaca dan gempa bumi.

2. Observasi

Metode yang digunakan untuk memperoleh data dengan cara mengadakan pengamatan terhadap objek penelitian dan pencatatan secara sistematis terhadap suatu gagasan yang diselidiki. Kegiatan yang dilakukan adalah melakukan riset untuk mengamati secara langsung proses penyampaian informasi cuaca dan gempa bumi pada BMKG Semarang.

3. Studi Pustaka

Merupakan teknik pencarian dengan melakukan pencarian data lewat literature-literatur yang terkait misalnya buku-buku referensi, artikel, materi diklat dan lain-lain seperti meminjam buku referensi masalah pemograman Delphi dan XML dari perpustakaan kampus Universitas Stikubank Semarang.

1.4.4. Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem adalah Waterfall (Whitten, 2004). Tahap yang digunakan dalam penelitian ini adalah:

1. Permulaan Sistem

Pada tahap ini menentukan jenis sistem yang tepat dan dapat menjawab persoalan yang dihadapi dalam pengembangan aplikasi aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway

2. Analisis Sistem

Pada tahap ini kegiatan yang dilakukan adalah mengenali dan mendefinisikan masalah pengembangan aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway serta menentukan prioritas penanganan masalah tersebut.

3. Perancangan Sistem

Pada tahap ini menyiapkan dan menyusun sistem baru, kemudian mengembangkan secara tertulis. Adapun kegiatan yang dilakukan meliputi merancang sistem dengan menggunakan OOD dan desain program.

4. Implementasi Sistem

Pada tahap ini melakukan implementasi aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway yang telah dibuat dan membuat sistem tersebut ke dalam program Delphi.

1.5. Sistematika Penulisan

Untuk memudahkan pemahaman tentang penelitian ini, maka pembahasan akan dibagi dalam beberapa bab sesuai dengan pokok permasalahannya, yaitu:

BAB I. PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Berisi tentang informasi hasil penelitian yang telah dilakukan sebelumnya dan menghubungkan dengan masalah yang sedang diteliti.

BAB III LANDASAN TEORI

Berisi tentang pembahasan atau penjelasan dari landasan teori dalam penelitian, seperti SMS gateway, OOD, Delphi.

BAB IV ANALISA DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai analisa dan perancangan aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway.

BAB V IMPLEMENTASI SISTEM

Pada Bab ini akan tahapan–tahapan pengembangan sistem dan disertakan implementasi yang memberikan gambaran tentang program yang dibuat dengan menggunakan Delphi.

BAB VI HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini berisi tentang hasil penelitian dan pembahasan.

BAB VII PENUTUP

Pada bab ini berisi tentang kesimpulan dan saran pembuatan sistem.

BAB II

TINJAUAN PUSTAKA

Pada bagian ini akan dipaparkan penjelasan tentang tinjauan pustaka yang dipakai dalam pembuatan aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway. Tinjauan Pustaka tersebut adalah hasil penelitian terdahulu tentang informasi hasil penelitian yang telah dilakukan sebelumnya dan menghubungkan dengan masalah yang sedang diteliti.

2.1. Pustaka Yang Terkait Dengan Penelitian

Hasil penelitian terdahulu dilakukan oleh Tejakusuma (2008) yaitu tentang “**Analisis Pasca Bencana Tsunami Ciamis-Cilacap**”. Penelitian ini membahas tentang gempa bumi yang terjadi akibat terjadinya sesar naik pada batas lempeng Australia dan Selat Sunda. Lempeng Australia bergerak ke arah utara–timur laut dengan kecepatan 59 mm/tahun. Lempeng Australia menunjam di bawah lempeng Sunda dan semakin dalam ke arah Pulau Jawa. Gempa bumi terjadi pada bagian yang dangkal, sekitar 50 kilometer dari palung Jawa ke arah utara. Daerah pada batas lempeng Sunda dan Australia merupakan daerah yang aktif. Beberapa gempa telah terjadi dan gempa yang terjadi di daerah ini seperti pada 2 Juni 1994 dengan skala Richter 7,8 yang menimbulkan tsunami dengan ketinggian gelombang mencapai 13 meter dan menewaskan sekitar 200 orang. Gempa tersebut terjadi di selatan Jawa sekitar

600 kilometer sebelah timur Ciamis. Pada 20 Agustus 1977, gempa berkekuatan 8,3 terjadi pada lempeng Australia sekitar 1200 kilometer sebelah timur – tenggara Ciamis. Gempa ini menimbulkan tsunami dengan tinggi gelombang mencapai 15 meter dan menewaskan sekitar 200 orang. Pada tahun 2006, tanggal 26 Mei, gempa besar telah mengguncang Yogyakarta dengan skala Richter 6,3 yang terjadi pada kedalaman dangkal namun tidak menimbulkan tsunami karena gempa terjadi di daratan. Gempa dengan kekuatan di atas 6,5 pada skala Richter dan terjadi di lautan sangat berpotensi menimbulkan tsunami dan terbukti pada kejadian gempa dan tsunami di wilayah Ciamis – Cilacap, khususnya Pangandaran dengan jumlah penduduk yang terbanyak. Menurut informasi dari masyarakat setempat, kejadian tsunami akibat gempa 17 Juli 2006 di selatan Ciamis terjadi sekitar 15-20 menit setelah kejadian gempa utama. Untuk mengetahui inundasi tsunami serta kerusakan yang ditimbulkannya maka dilakukan penelitian di daerah terdampak yaitu Ciamis – Cilacap. Hasil dari penelitian ini adalah teori yang menyebutkan bahwa gempa dengan kekuatan di atas 6,3 skala Richter dan terjadi di lautan serta dengan kedalaman pusat gempa di bawah 30 kilometer telah terbukti benar. Hal ini ditunjukkan oleh gempa yang terjadi pada 17 Juli 2006 di lepas pantai selatan Ciamis dengan kekuatan 7,7 skala Richter dengan kedalaman gempa 10 kilometer. Gempa ini telah menimbulkan gelombang tsunami yang menyapu daerah pesisir pantai Ciamis – Cilacap dengan ketinggian gelombang beberapa meter hingga tidak kurang dari 7.9 meter. Tak kurang dari 500 orang

korban jiwa dan ratusan rumah hancur akibat tsunami ini. Bangunan dengan konstruksi kayu di bibir pantai tersapu habis oleh gelombang tsunami sementara bangunan dengan konstruksi beton terlihat lebih kuat. Bangunan dengan konstruksi bata tanpa penguatan tulang beton terlihat tersapu habis oleh gelombang tsunami.

Penelitian yang dilakukan oleh Budiarto dkk (2012) tentang **“Sistem Monitoring Tingkat Ketinggian Air Bendungan Berbasis Mikrokontroler”** yang menjelaskan tentang sistem peringatan dini berupa sistem peringatan dini bencana secara real time (Real Time Hazard Early Warning System) yang bertujuan untuk memberikan informasi bencana terhadap masyarakat /pihak terkait sehingga nantinya dapat mempersiapkan diri dan meminimalisir korban jiwa dengan memberikan informasi sedini mungkin akan adanya bencana yang mungkin bisa terjadi. Sensor deteksi dini untuk mengukur ketinggian debit air bendungan untuk mendeteksi debit air sungai di bendungan yang melewati pintu air dengan ketinggian yang memungkinkan bisa terjadinya banjir. Pemerintah Kota Semarang senantiasa melakukan pembenahan berkala berkaitan dengan penanggulangan banjir bagi segenap masyarakat daerah yang berpotensi terhadap banjir dan dampaknya secara meluas. Salah satunya adalah pengadaan alat pendeteksi banjir yang ditempatkan pada lokasi bendungan. Dalam penelitian ini digunakan mikrokontroler ATmega 8535, pemrograman menggunakan CVAVR, sedangkan perangkat lunak pemrograman dan donloader menggunakan

CodeVisionAVR Versi 2.03.9. Program interface yang digunakan terdiri dari program untuk deteksi sensor, program untuk pengolahan sinyal dan program untuk menghasilkan informasi yang ditampilkan pada LCD dan LED. Hasil yang diperoleh dalam penelitian adalah dalam perancangan perangkat keras diperlukan ketelitian dalam menghitung besar kecilnya tegangan dan arus yang dihasilkan oleh sensor, karena besarnya tegangan yang dihasilkan oleh sensor akan berpengaruh pada kinerja mikrokontroler. Hasil yang diperoleh dari peralatan ini masih bersifat kualitatif. Agar peralatan dapat berfungsi secara optimal diperlukan sensor yang dapat menghasilkan besaran sinyal secara kuantitatif. Sehingga perubahan tingkat ketinggian dapat diamati lebih teliti.

Penelitian yang dilakukan oleh Rosely (2011) tentang **“Aplikasi Informasi Cuaca Dan Alert Bencana Alam Berbasis Sms Gateway”**. Penelitian ini menjelaskan tentang BMKG Makasar atau sering disebut dengan Badan Meteorologi, Klimatologi dan Geofisika Makasar merupakan sebuah instansi yang menangani tentang masalah gempa bumi, cuaca, iklim dan tsunami. Selama ini BMKG dalam menangani masalah penyampaian informasi tentang gempa bumi dan tsunami masih dengan cara turun langsung ke lapangan, yaitu dengan cara memberikan informasi tersebut dengan mendatangi daerah-daerah yang terjadi gempa bumi dan kemungkinan akan terjadi tsunami. Proyek akhir ini dibangun dalam beberapa tahap, yaitu tahap analisis, desain, *coding*, pengujian, dan penerapan aplikasi. Desain aplikasi menggunakan desain system dengan Data Flow Diagram. Sedangkan pada tahap *coding*,

aplikasi ini dibangun menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dengan database MySQL. Pengguna aplikasi ini adalah berbagai user yang ingin mengetahui mengenai cuaca dan bencana alam yang terjadi belakangan ini. Aplikasi ini mempunyai fungsi yaitu pemberitahuan mengenai bencana alam yang terjadi dan membantu masyarakat yang ingin mengetahui mengenai keadaan cuaca daerahnya. Dengan adanya aplikasi ini diharapkan dapat mempermudah pengguna untuk mengetahui mengenai cuaca dan bencana alam.

2.2. Perbedaan Penelitian Yang Dilakukan Dengan Penelitian Terdahulu

Perbedaan dari penelitian yang penulis buat yaitu penulis menggunakan Delphi dan XML dalam pembuatan aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway yang digunakan untuk menampilkan informasi cuaca dan gempa bumi yang bersumber dari website www.bmkgjateng.com. Untuk membuat tampilan lebih menarik, didalam aplikasi yang dibuat menambahkan pemrograman Delphi dengan skin yang dapat diubah-ubah yang penerapannya untuk mempercantik tampilan aplikasi yang dibuat.

BAB III

LANDASAN TEORI

3.1. Cuaca

Cuaca merupakan keadaan atmosfer pada suatu tempat dan waktu tertentu, biasanya diperhitungkan pada kondisi harian. Cuaca terbentuk dari gabungan unsur cuaca dan jangka waktu cuaca bisa hanya beberapa jam saja, misalnya: pagi hari, siang hari atau sore hari, dan keadaannya bisa berbeda-beda untuk setiap tempat serta setiap jamnya. Di Indonesia keadaan cuaca selalu diumumkan untuk jangka waktu sekitar 24 jam melalui prakiraan cuaca hasil analisis Badan Meteorologi Klimatologi dan Geofisika (BMKG). Misalnya, Badan Meteorologi Klimatologi dan Geofisika memperkirakan cuaca Jakarta esok hari cerah, dengan suhu rata-rata maksimum 31°C dan suhu minimumnya 24°C . Umumnya kajian cuaca hanya meliputi temperatur, curah hujan, dan angin. (<http://www.bmkg.go.id>)

Unsur-unsur cuaca terdiri dari temperatur, angin, tekanan udara, curah hujan, dan kelembapan udara.

1. Temperatur

Temperatur merupakan derajat panas akibat aktivitas molekul dalam atmosfer. Biasanya temperatur pengukur dinyatakan dalam skala Celcius, Reamur, dan Fahrenheit. Namun temperatur kondisi suhu di muka bumi tidaklah sama di beberapa tempat.

2. Angin

Angin merupakan udara yang bergerak dari daerah bertekanan udara tinggi ke daerah bertekanan udara rendah. Angin ini diakibatkan oleh temperatur atau suhu yang dihasilkan oleh matahari sehingga muncullah angin. Kecepatan gerakan angin atau kecepatan angin biasanya diukur berdasarkan skala tertentu, seperti salah satunya skala kecepatan angin yaitu skala Beaufort. Jendral Beaufort merupakan penemu skala ini pada awal abad 19 dan di gunakan di seluruh dunia (internasional).

3. Tekanan Udara

Tekanan udara merupakan gaya yang diakibatkan oleh kolom udara secara vertikal di bidang yang luasnya 1 cm². Tekanan udara diukur menggunakan barometer dan dinyatakan dengan milibar (mb).

4. Curah Hujan

Curah hujan merupakan jumlah air yang jatuh di permukaan tanah selama kurun waktu tertentu yang diukur dengan satuan tinggi (mm) di atas permukaan horisontal bila tidak terjadi infiltrasi, evaporasi dan *run off*. Satuan CH adalah mm, inch.

5. Kelembaban Udara

Kelembaban udara adalah banyaknya uap air yang terkandung dalam udara pada tempat tertentu dan periode tertentu. Alat yang digunakan dalam mengukur kelembapan udara yaitu *hygrometer* dan *psychrometer*. Kelembaban udara bisa dibedakan menjadi dua yaitu kelembaban nisbi dan kelembaban mutlak

3.2. Gempa Bumi

Gempa bumi adalah peristiwa bergetarnya bumi akibat pelepasan energi di dalam bumi secara tiba-tiba yang ditandai dengan patahnya lapisan batuan pada kerak bumi. Akumulasi energi penyebab terjadinya gempa bumi dihasilkan dari pergerakan lempeng-lempeng tektonik. Energi yang dihasilkan dipancarkan kesegala arah berupa gelombang gempa bumi sehingga efeknya dapat dirasakan sampai ke permukaan bumi. (<http://www.bmkg.go.id>)

Gempa bumi biasa disebabkan oleh pergerakan kerak bumi (lempeng bumi). Frekuensi suatu wilayah, mengacu pada jenis dan ukuran gempa bumi yang di alami selama periode waktu. Gempa bumi diukur dengan menggunakan alat Seismometer. Moment magnitudo adalah skala yang paling umum di mana gempa bumi terjadi untuk seluruh dunia. Skala Rickter adalah skala yang di laporkan oleh observatorium seismologi nasional yang di ukur pada skala besarnya lokal 5 magnitudo. kedua skala yang sama selama rentang angka mereka valid. gempa 3 magnitudo atau lebih sebagian besar hampir tidak terlihat dan besar nya 7 lebih berpotensi menyebabkan kerusakan serius di daerah yang luas, tergantung pada kedalaman gempa. Gempa bumi terbesar bersejarah besarnya telah lebih dari 9, meskipun tidak ada batasan besarnya. Gempa bumi besar terakhir besarnya 9,0 atau lebih besar adalah 9,0 magnitudo gempa di Jepang pada tahun 2011 (per Maret 2011), dan itu adalah gempa Jepang terbesar sejak pencatatan dimulai. Intensitas getaran diukur pada modifikasi Skala Mercalli.

3.3. SMS Gateway

3.3.1. SMS (*Short Message Service*)

SMS adalah salah satu fasilitas yang terdapat pada ponsel yang memiliki fungsi untuk mengirimkan paket pesan singkat yang berupa tulisan kata. SMS dibagi menjadi 3 kategori yaitu *Plain SMS*, *Encoded SMS* dan *Encrypted SMS*. *Plain SMS* adalah aplikasi dimana komunikasi antara dua pihak yang berhubungan menggunakan teks biasa. *Encoded SMS* adalah aplikasi yang menggunakan SMS dengan sebuah format tertentu yang dikenali oleh ponsel yang bersangkutan seperti ringtone, operator logo, picture messaging, screen saver, startup logo dan lain-lain. *Encrypted SMS* adalah mengirimkan dan menerima SMS dalam format yang terenkripsi sehingga aman untuk digunakan bertransaksi. Enkripsi adalah sebuah metode yang mengubah data yang seharusnya menjadi rangkaian kode sehingga sulit untuk dipecahkan. Aplikasi seperti *Mobile Banking* yang membutuhkan tingkat keamanan yang tinggi memerlukan metode ini untuk operasinya. (Cahyanto, 2003:1)

3.3.2. SMS Gateway

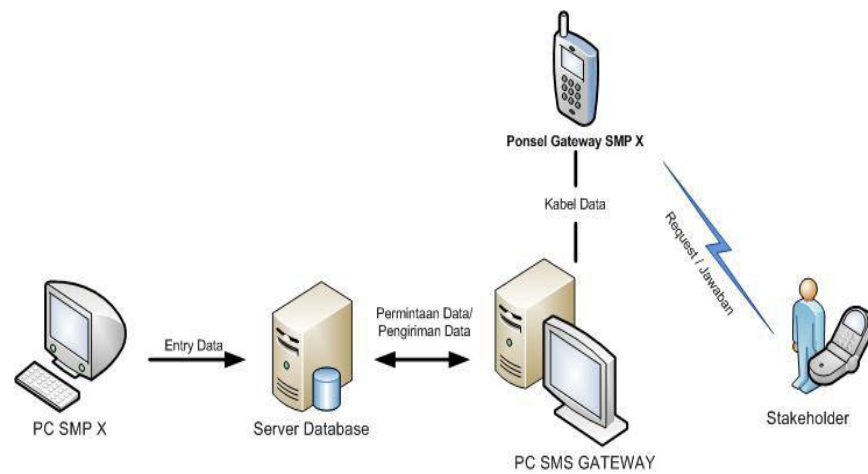
SMS Gateway adalah suatu sistem yang menjembatani antara handphone dengan sistem yang menjadi server dengan SMS sebagai informasinya. *SMS Gateway* tidak memerlukan koneksi internet manapun karena sifatnya memang bekerja sendirian (*stand alone*). *SMS Gateway* memerlukan satu atau beberapa buah terminal. Pemilihan

banyak terminal akan menjadikan pengiriman dan penerimaan SMS semakin cepat.

Pada SMS Gateway, data yang dikirimkan ke telepon seluler peminta sebagai respon atas permintaan tersebut. Data-data yang disediakan oleh penyedia data dikelompokkan dengan kode-kode tertentu yang sudah distandarkan dan sudah terbentuk format tertentu yang disesuaikan dengan kemampuan SMS. Jadi peminta dapat memilih data mana yang diinginkan dengan mengirimkan kode tertentu yang sudah distandarkan tadi.

3.3.3. Komponen Pendukung *SMS Gateway*

1. PC atau laptop (sms server) yang digunakan untuk meletakkan aplikasi SMS Gateway dan administrasi yang akan dibangun.
2. Handphone (*Receiver*) yang digunakan untuk menerima sms dalam hal ini sms yang telah dikirim oleh handphone pemakai (originator).
3. Handphone (Originatur) yang digunakan untuk mengirimkan sms ke handphone (*receiver*).
4. Kabel data yang digunakan untuk pentransferan sms dari handphone (*receiver*) ke PC atau laptop dan sebaliknya. Bila menghubungkan ponsel melalui port serial tidak akan dapat terdeteksi secara otomatis oleh komputer. Jika menggunakan kabel data kestabilan koneksinya tidak akan terputus selama kabelnya tidak dicabut.



Gambar 3.1. Alur SMS Gateway

3.3.4. Keuntungan SMS Gateway

1. Dapat menyebarkan pesan ke ratusan nomor secara otomatis dan cepat yang langsung terhubung dengan database nomor-nomor ponsel saja tanpa harus mengetik ratusan nomor dan pesan di ponsel karena semua nomor akan diambil secara otomatis dari database tersebut
2. Dapat mengotomisasi pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita karena biasanya pesan yang ingin dikirim berbeda-beda untuk masing-masing penerimanya.

3.3.5. Kekurangan SMS Gateway

1. Jika terdapat gangguan pada jaringan telekomunikasi, maka sistem tidak dapat secara otomatis mengirim ulang pesan yang telah dikirimkan.

2. Tidak mendukung pengiriman sms dalam bentuk gambar dan suara melainkan dalam bentuk teks saja.

3.3.6. Model SMS Gateway

Beberapa fitur atau model yang umum dikembangkan dalam aplikasi SMS gateway adalah (Budicahyanto, 2003:15):

1. *Auto Reply*

SMS gateway secara otomatis akan membalas SMS yang masuk. Pengirim mengirimkan SMS dengan format tertentu yang dikenali aplikasi, kemudian aplikasi dapat melakukan *auto-reply* dengan membalas SMS tersebut, berisi informasi yang dibutuhkan.

2. Pengiriman Masal

Disebut juga dengan istilah SMS *broadcast* atau jenis sms satu arah ke banyak nomor tujuan yang bertujuan untuk mengirimkan SMS ke banyak tujuan sekaligus.

3. Pengiriman Terjadwal

Sebuah SMS dapat diatur untuk dikirimkan ke tujuan secara otomatis pada waktu tertentu. Contohnya untuk keperluan mengucapkan selamat ulang tahun.

3.4. Desain Berorientasi Objek

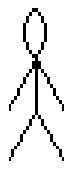

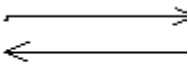
Desain Berorientasi Objek (*Objek Oriented Design*) adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem (Gunadi, 2002). Desain Berorientasi Objek

sebagai tahap lanjutan setelah Analisis Berorientasi Objek dimana tujuan sistem diorganisasi ke dalam subsistem berdasar struktur analisis dan arsitektur yang dibutuhkan.

3.4.1. *Use Case*

Use case diagram adalah sebuah diagram yang menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. (Gunadi, 2002) Komponen atau simbol yang digunakan dalam use case diagram meliputi :

Tabel 3.1. Simbol *Use Case*
(Gunadi, 2002)

No.	Komponen	Arti	Keterangan
1.		Actor	Manusia, user, pengguna sistem yang berhubungan secara langsung dengan sistem
2.		Proses	Perilaku yang ditunjukkan atau dilakukan oleh actor.
3.		Relasi	Penghubung antara actor dengan proses atau proses dengan proses.

Dalam relasi ini terdapat dua komponen yaitu :

a. *Include*

Adalah suatu bagian dari elemen (yang ada di garis tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah).

b. *Extend*

Adalah menunjukkan suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah.

3.4.2. Activity Diagram

Activity diagram menggambarkan alur kerja sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah flowchart karena dapat memodelkan sebuah alur kerja dari suatu aktifitas ke aktifitas lainnya atau dari satu aktifitas kedalam keadaan sesaat (state). Activity diagram bermanfaat untuk memahami proses secara keseluruhan.










Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Activity diagram menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision*

digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa object untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Tabel 3.2. Simbol *Activity Diagram*
(Gunadi, 2002)

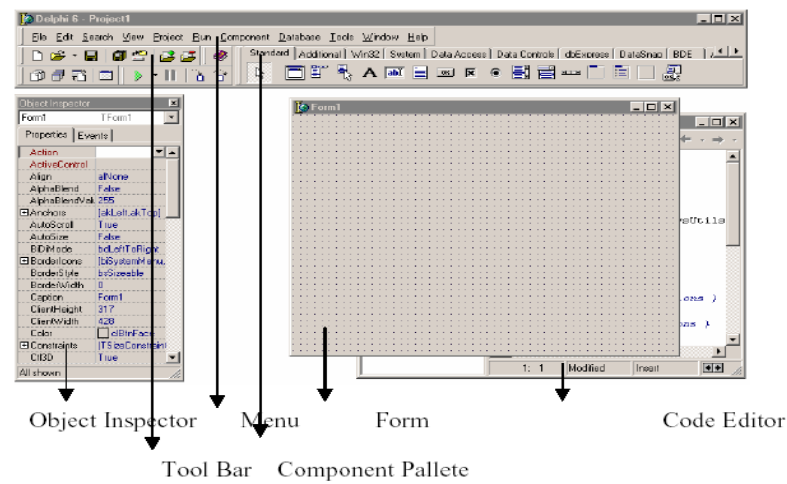
Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; Untuk menunjukkan kegiatan yang dilakukan secara paralel
	Rake; menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Penerimaan
	Aliran Akhir (Flow Final)

3.5. Delphi

Delphi adalah salah satu program secara visual, bahasa yang digunakan lebih mengarah pada bahasa pemrograman pascal.

3.5.1. Integrated Development Environment

Tampilan bidang kerja yang disebut dengan IDE (*Integrated Development Environment*) Delphi bisa dilihat pada gambar 3.2. IDE ini secara garis besar terdiri atas tiga bagian utama, yaitu window utama, object inspector dan editor.



Gambar 3.2. Tampilan IDE Delphi

Window Utama terdiri atas menu bar, tool bar dan component palette. Object inspector menyediakan dua kelompok pengaturan, yaitu properties dan events. Editor yang disediakan ada dua buah yaitu form editor dan code editor.

3.5.2. Menu Bar

Menu bar menyediakan kelompok perintah yang digolongkan dalam sebelas menu. Menu tersebut antara lain file, edit, search, view, project, run, component, database, tools, window, dan help.



Gambar 3.3 Menu Bar

Komponen menu bar :

1. File

Kelompok perintah yang berfungsi untuk pengaturan suatu file.

Misalnya new, save, open dan sebagainya.

2. Edit

Menyimpan perintah-perintah untuk pengeditan. Mulai dari pengeditan objek, pengeditan komponen maupun pengeditan kode pada code editor. Contohnya cut, paste, align.

3. Search

Kelompok perintah yang berfungsi untuk melakukan proses pencarian, baik objek, kata, kalimat dan sebagainya.

4. View

Menampung perintah-perintah untuk mengaktifkan bagian pendukung *integrated development environment*.

5. Project

Kelompok perintah yang berfungsi untuk manajemen proyek berikut bagian-bagian pendukungnya.

6. Run

Kelompok perintah untuk menangani proses kompilasi program seperti run, parameters, inspect dan seterusnya.

7. Component

Mengatur suatu komponen.

8. Database

Mengandung perintah-perintah untuk pengaturan aplikasi database.

9. Tools

Kelompok perintah yang berfungsi sebagai penyedia perlengkapan tambahan yang diperlukan dalam penyusunan program seperti image editor, database desktop, dan lain-lain.

10. Window

Melihat jendela aplikasi yang ada.

11. Help

Memberikan informasi yang bersifat menolong pemakai dalam menggunakan delphi.

3.5.3. Tool Bar

Tool bar adalah bagian dari delphi yang menyediakan tombol-tombol speed. Tombol speed fungsinya sama dengan perintah yang ada pada menu. Tombol ini disediakan dengan tujuan meringkas atau mempercepat pekerjaan kita. Untuk mengakses suatu perintah dengan menggunakan menu, langkah yang kita lakukan adalah klik menu lalu klik submenu kemudian klik perintah tersebut. Sedangkan untuk mengakses perintah dengan menggunakan tombol speed, langkah yang kita lakukan hanya mengklik tombol yang kita inginkan.



Gambar 3.4. Tool Bar

3.5.4. Component Palette

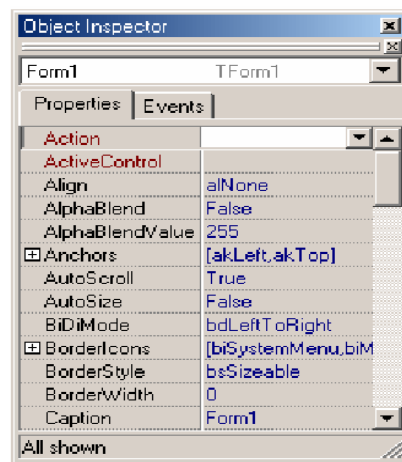
Component palette menyediakan berbagai komponen yang bisa kita pasang pada form sesuai keperluan kita. Telah tersedia berbagai komponen yang dikelompokkan seperti pada gambar 3.5.



Gambar 3.5. Component Palette

3.5.5. Object Inspector

Object inspector adalah sarana pengaturan objek yang kita pasang pada form, atau form itu sendiri. Dua hal penting yang bisa kita setel pada komponen adalah Properti dan Event. Properties adalah yang terkait dengan sifat komponen seperti ukuran, warna dan sebagainya. Sedangkan event adalah kejadian atau peristiwa yang kita inginkan terpasang pada komponen tersebut kaitannya dengan proses pemakaian. Contoh event misalnya klik, klik ganda, drag (geser), drop dan sebagainya.



Gambar 3.6. Object Inspector

3.5.6. Form

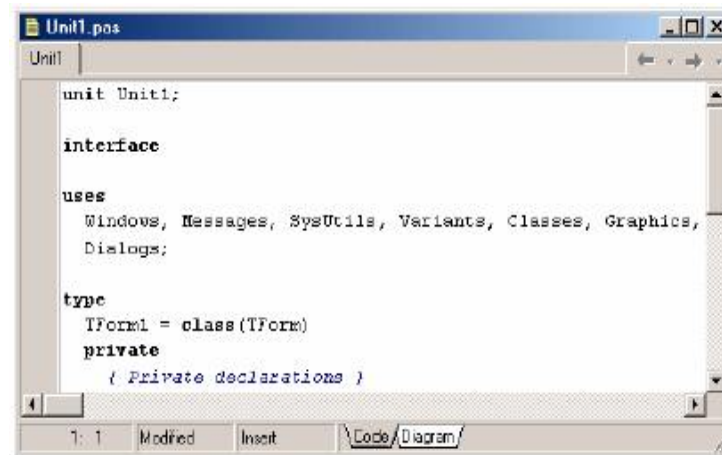
Form adalah bahan dasar yang akan menjadi jendela aplikasi kita. Pada form terdapat tiga tombol kontrol, yaitu minimize, maximize atau restore dan close. Terdapat juga *caption bar* tempat kita menempatkan judul form (yang kelak menjadi judul window) dan icon. Pembatas form juga bisa diubah ukurannya dengan cara *drag* (geser) *and drop*. Pada form kita bisa meletakkan komponen-komponen yang kita perlukan dalam suatu *user interface*.



Gambar 3.7. Form

3.5.7. Code Editor

Code editor adalah tempat kita menuliskan program dalam bahasa object pascal. Secara default code editor ini terletak di belakang form editor.



Gambar 3.8. Code Editor

3.5.8. Tipe Data Pada Delphi

Tipe data yang digunakan dalam pemrograman delphi adalah

1. String

Digunakan untuk masukan data string (string merupakan gabungan beberapa karakter).

2. Integer

Merupakan bilangan asli (1, 2, 3, 4, ...).

2. Real

Bilangan bulat (berupa pecahan dan integer).

3. Byte

Besar dari tipe ini hanya 8 bit.

4. Word

Besar bilangan yang terdiri dari 16 bit.

5. Date and time

Tipe waktu dan tanggal yang disediakan delphi

3.5.9. Konversi Tipe Data

Dalam pengolahan data memerlukan suatu konversi tipe data.

Contoh konversi tipe data yang dapat dilakukan oleh delphi antara lain :

1. Strtoint

Mengubah tipe data string ke integer.

2. Inttostr

Mengubah tipe data integer ke string.

3. Timetostr

Mengubah tipe data time ke dalam bentuk string.

4. Strtfloat

Mengubah tipe data string ke dalam bentuk real.

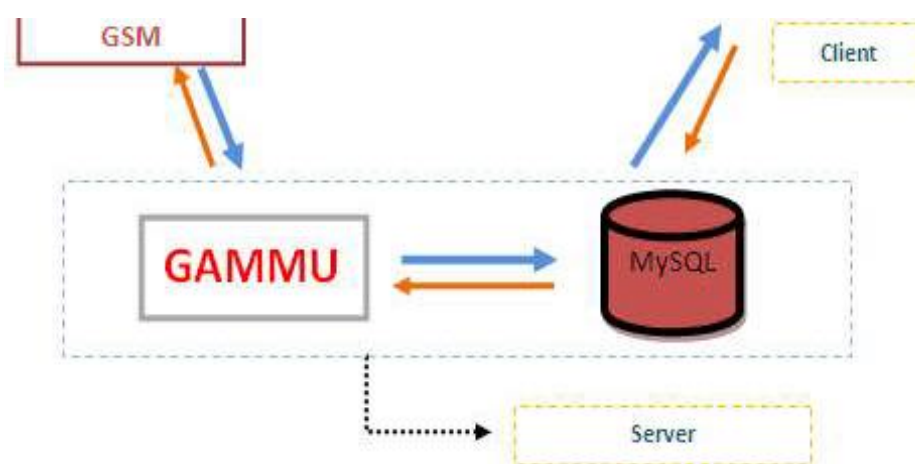
5. Floattostr

Mengubah tipe data real ke dalam bentuk string.

3.6. Gammu

Gammu adalah sebuah aplikasi/daemon yang dikhususkan untuk membangun sebuah SMS Gateway yang menghubungkan antara operator seluler ke internet dan sebaliknya. Aplikasi ini bersifat *open source* dibawah lisensi GPL. Sebenarnya Gammu tidak hanya berfungsi sebagai SMS Gateway

saja tetap bisa juga untuk melakukan *voice call*. Saat ini gammu telah menyediakan codebase yang stabil dan mapan untuk berbagai macam model telepon yang tersedia di pasaran dibandingkan dengan project sejenis. Gammu merupakan project yang berlisensi GNU GPL 2 sehingga menjamin kebebasan menggunakan tool ini tanpa perlu takut dengan masaah legalitas dan biaya yang mahal yang harus dikeluarkan. Gammu mendukung berbagai macam model telepon seluler dengan berbagai jenis koneksi dan tipe. (<http://wammu.eu/gammu/>).



Gambar 3.9. Gammu

1. GSM

Sebagai perangkat penerima sms dan pengirim sms. Handphone-handphone keluaran lama maupun baru dapat digunakan sebagai modem GSM. Masing-masing memiliki spesifikasi dan memiliki tipe koneksi yang berbeda-beda karena tipe koneksi berguna sekali untuk mengkoneksikan handphone/modem GSM ke perangkat komputer.

2. Gammu

Sebagai perangkat penerjemah interkoneksi protokol handphone agar bisa digunakan datanya pada perangkat komputer. Penjelasan ini bisa diartikan sebagai “Gateway”. Gammu menguasai setiap protokol-protokol atau fungsifungsi pada handphone, sehingga perangkat komputer dapat membaca tipe data dari handphone tersebut. Koneksi dari handphone menuju perangkat komputer menggunakan berbagai jenis perangkat. Menggunakan Kabel data, Infra Red atau bahkan menggunakan Bluetooth.

3. MySQL

Merupakan perangkat back-end dari perangkat komputer. Data dari handphone atau modem GSM setelah “dikuasai” dan “diterjemahkan” oleh Gammu, diletakkan atau diarahkan menuju database, Gammu menggunakan MySQL sebagai Database. Ini yang membuat perangkat komputer selanjutnya dapat mengolah data mentah SMS dari MySQL.

4. Server

Perangkat Gammu dan MySQL server di letakkan pada perangkat komputer yang selanjutnya disebut Server atau SMS Gateway Server. Penggunaan Server ini dapat dikembangkan selanjutnya sesuai dengan keinginan pengguna menggunakan data yang tersedia pada MySQL.

5. Client

Perangkat yang meng-implementasikan aplikasi yang telah dibangun untuk mengolah data sms dari modem GSM. Aplikasi inilah yang disebut dengan Aplikasi SMS sehingga dapat berkirim SMS dan menerima

SMS dari perangkat Client. Jadi seolah-olah client merupakan perangkat handphone yang fungsinya bisa menerima dan mengirim SMS.

3.7. XML

XML (*eXtensible Markup Language*) merupakan bahasa web turunan dari SGML (*Standart Generalized Markup Language*) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapatkan pengakuan dari *World Wide Web Consortium* (W3C) pada bulan Februari 1998. Sedangkan SGML sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa markup yang tidak kalah hebatnya dengan SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa markup adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standart dan vendor-independent.

XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa markup lain, sehingga dinamakan meta-language. Meskipun demikian pendapat yang XML bukan merupakan bahasa markup, didasarkan bahwa XML merupakan bahasa markup terpisah untuk tujuan terpisah. Selain itu XML bukanlah solusi semua hal untuk tujuan semua user. Sedangkan peran dari markup itu sendiri berupa:

1. Markup dapat menambah maksud arti (semantic) suatu data.
2. Dapat memisahkan data.
3. Dapat mendefinisikan peran data.
4. Dapat mendefinisikan batasan data.
5. Dapat mendefinisikan keterhubungan data.

XML merupakan sebuah himpunan bagian (subset) dari SGML yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses di dalam web dengan cara seperti yang dimungkinkan HTML saat ini. XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa markup yang dirancang untuk penyampaian informasi melalui *World Wide Web* (www) atau sering disebut web saja.

3.8. MySQL

MySQL adalah relational database management system (RDBMS) yang didistribusikan secara gratis dibawah licensi GPL (*General Public License*). MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database terutama untuk pemilihan/seleksi dan pemasukan data yng memungkinkan pengoperasian data dikerjakan dengan mudah dan secara otomatis. (Prasetyo, 2003)

Keandalan suatu sistem database dapat diketahui dari cara kerja optimizer nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query My SQL dapat sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

3.8.1. Keistimewaan MySQL

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan (Prasetyo, 2003) diantaranya :

1. *Portability*

MySQL dapat berjalan stabnil pada berbagai sistem operasi diantaranya seperti Windows, Linux, FreeBSD, Mac OS X Server dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan secara *open source* (gratis) dibawah licensi GPL.

3. *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah database server MySQL dapat diakses clieny secara bersamaan.

4. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned* integer, float, double, char dan masih banyak lagi.

6. *Command dan Function*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan ijin akses user dengan sistem perijinan yang mendetail serta password terenkripsi.

8. *Scalability dan Limits*

MySQL mampu menangani database dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks per tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket (Unix) atau Named Pipes (NT).

10. *Localisation*

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa.

11. *Interface*

MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API.

12. *Client dan Tools*

MySQL dilengkapi dengan berbagai tools yang dapat digunakan untuk administrasi database dan pada setiap tool yang ada disertakan petunjuk online

13. *Struktur Tabel*

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

BAB IV

ANALISA DAN PERANCANGAN SISTEM

4.1. Gambaran Umum Perusahaan

4.1.1. Sejarah Perusahaan

Sejarah pengamatan meteorologi dan geofisika di Indonesia dimulai pada tahun 1841 diawali dengan pengamatan yang dilakukan secara perorangan oleh Dr. Onnen, Kepala Rumah Sakit di Bogor. Tahun demi tahun kegiatannya berkembang sesuai dengan semakin diperlukannya data hasil pengamatan cuaca dan geofisika. Pada tahun 1866, kegiatan pengamatan perorangan tersebut oleh pemerintah Hindia Belanda diresmikan menjadi instansi pemerintah dengan nama *Magnetisch en Meteorologisch Observatorium* atau Observatorium Magnetik dan Meteorologi dipimpin oleh Dr. Bergsma. Pada tahun 1879 dibangun jaringan penakar hujan sebanyak 74 stasiun pengamatan di Jawa. Pada tahun 1902 pengamatan medan magnet bumi dipindahkan dari Jakarta ke Bogor.

Pengamatan gempa bumi dimulai pada tahun 1908 dengan pemasangan komponen horisontal seismograf Wiechert di Jakarta, sedangkan pemasangan komponen vertikal dilaksanakan pada tahun 1928. Pada tahun 1912 dilakukan reorganisasi pengamatan meteorologi dengan menambah jaringan sekunder. Sedangkan jasa meteorologi mulai digunakan untuk penerangan pada tahun 1930.

Pada masa pendudukan Jepang antara tahun 1942 sampai dengan 1945, nama instansi meteorologi dan geofisika diganti menjadi *Kisho Kauso Kusho*. Setelah proklamasi kemerdekaan Indonesia pada tahun 1945, instansi tersebut dipecah menjadi dua yaitu di Yogyakarta bentuk biro meteorologi yang berada di lingkungan Markas Tertinggi Tentara Rakyat Indonesia khusus untuk melayani kepentingan Angkatan Udara dan di Jakarta dibentuk jawatan Meteorologi dan Geofisika, dibawah Kementerian Pekerjaan Umum dan Tenaga Pada tanggal 21 Juli 1947 Jawatan Meteorologi dan Geofisika diambil alih oleh Pemerintah Belanda dan namanya diganti menjadi *Meteorologisch en Geofysische Dienst*.

Pada tahun 1949, setelah penyerahan kedaulatan negara Republik Indonesia dari Belanda, *Meteorologisch en Geofysische Dienst* diubah menjadi jawatan Meteorologi dan Geofisika dibawah Departemen Perhubungan dan Pekerjaan Umum. Selanjutnya, pada tahun 1950 Indonesia secara resmi masuk sebagai anggota Organisasi Meteorologi Dunia (*World Meteorological Organization* atau WMO) dan Kepala Jawatan Meteorologi dan Geofisika menjadi *Permanent Representative of Indonesia with WMO*. Pada tahun 1955 Jawatan Meteorologi dan Geofisika diubah namanya menjadi Lembaga Meteorologi dan Geofisika di bawah Departemen Perhubungan, dan pada tahun 1960 namanya dikembalikan menjadi Jawatan Meteorologi dan Geofisika di bawah Departemen

Perhubungan Udara. Pada tahun 1965, namanya diubah menjadi Direktorat Meteorologi dan Geofisika, kedudukannya tetap di bawah Departemen Perhubungan Udara. Pada tahun 1972, Direktorat Meteorologi dan Geofisika diganti namanya menjadi Pusat Meteorologi dan Geofisika, suatu instansi setingkat eselon II di bawah Departemen Perhubungan, dan pada tahun 1980 statusnya dinaikkan menjadi suatu instansi setingkat eselon I dengan nama Badan Meteorologi dan Geofisika, tetap berada di bawah Departemen Perhubungan. Terakhir pada tahun 2002, dengan keputusan Presiden RI Nomor 46 dan 48 tahun 2002, struktur organisasinya diubah menjadi Lembaga Pemerintah Non Departemen (LPND) dengan nama tetap Badan Meteorologi dan Geofisika. Kemudian disesuaikan dengan Peraturan Pemerintah PP No. 61 Tanggal 2 September 2008 yang memutuskan Nama BMG dirubah menjadi Badan Meteorologi Klimatologi dan Geofisika Semarang.

4.1.2. Visi, Misi dan Tupoksi

1. Visi

Terwujudnya Badan Meteorologi Klimatologi dan Geofisika Semarang yang tanggap dan mampu memberikan pelayanan meteorologi, klimatologi, kualitas udara dan geofisika yang handal guna mendukung keselamatan dan keberhasilan pembangunan nasional serta berperan aktif di tingkat internasional.

2. Misi

- a. Mengamati dan memahami fenomena meteorologi, klimatologi, kualitas udara dan geofisika.
- b. Menyediakan data dan informasi meteorologi, klimatologi, kualitas udara dan geofisika yang handal dan terpercaya
- c. Melaksanakan dan mematuhi kewajiban internasional dalam bidang meteorologi, klimatologi, kualitas udara dan geofisika.
- d. Mengkoordinasikan dan memfasilitasi kegiatan di bidang meteorologi, klimatologi, kualitas udara dan geofisika.

3. Tupoksi

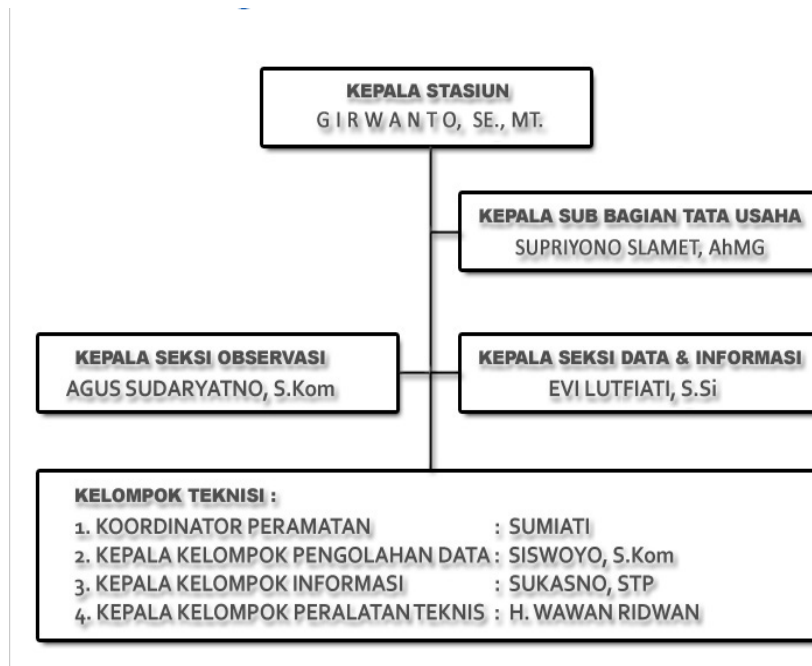
Badan Meteorologi Klimatologi dan Geofisika Semarang mempunyai status sebuah Lembaga Pemerintah Non Departemen (LPND) yang dipimpin oleh seorang kepala badan. Badan Meteorologi Klimatologi dan Geofisika Semarang mempunyai tugas yaitu melaksanakan tugas pemerintahan di bidang meteorologi, klimatologi, kualitas udara, dan geofisika sesuai dengan ketentuan perundang-undangan yang berlaku.

Dalam melaksanakan tugas sebagaimana dimaksud diatas, Badan Meteorologi Klimatologi dan Geofisika Semarang menyelenggarakan fungsi :

- a. Pengkajian dan penyusunan kebijakan nasional di bidang meteorologi, klimatologi, kualitas udara dan geofisika.

- b. Koordinasi kegiatan fungsional di bidang meteorologi, klimatologi, kualitas udara dan geofisika.
- c. Fasilitasi dan pembinaan terhadap kegiatan instansi pemerintah dan swasta di bidang meteorologi, klimatologi, kualitas udara dan geofisika.
- d. Penyelenggaraan pengamatan, pengumpulan dan penyebaran, pengolahan dan analisis serta pelayanan di bidang meteorologi, klimatologi, kualitas udara dan geofisika. Penyelenggaraan kegiatan kerjasama di bidang meteorologi, klimatologi, kualitas udara dan geofisika.
- e. Penyelenggaraan pembinaan dan pelayanan administrasi umum di bidang perencanaan umum, ketatausahaan, organisasi dan tatalaksana, kepegawaian, keuangan, kearsipan, hukum, persandian, perlengkapan dan rumah tangga.

4.1.3. Struktur Organisasi



Gambar 4.1. Struktur Organisasi

4.2. Analisa Sistem

Tahap analisa sistem adalah studi domain masalah untuk merekomendasikan perbaikan dan menspesifikasi persyaratan dan prioritas untuk solusi. Tugas paling penting dalam tahap ini adalah proses menemukan masalah dan menghasilkan alternatif pemecahan masalah serta diharapkan dapat memahami sistem yang ada guna menentukan kebutuhan pemakai dan hambatan pada sistem yang baru.

4.2.1 Analisa Permasalahan

BMKG atau sering disebut dengan Badan Meteorologi, Klimatologi dan Geofisika merupakan sebuah instansi yang menangani tentang masalah iklim, tsunami, informasi cuaca di kota

tertentu, informasi posisi gempa bumi terbaru, suhu udara, kelembaban, lama penyinaran matahari, tekanan udara dan lain sebagainya dalam kurun waktu tertentu maupun data sekunder seperti prakiraan cuaca esok hari, prakiraan cuaca seminggu kedepan, iklim dan lain-lain yang dapat digunakan untuk tujuan penelitian.

Perancangan aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* merupakan pemecahan dari masalah tersebut yang dapat mempermudah dan mempercepat pihak BMKG dalam menginformasikan tentang cuaca dan bencana alam yang terjadi kepada masyarakat sehingga dapat mengetahui informasi tentang cuaca dan gempa bumi.

4.3. Perancangan Sistem

4.3.1. Deskripsi Sistem

Aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* dapat digunakan oleh pengguna untuk mengirimkan sms informasi cuaca dan gempa bumi, jika pengguna mengirimkan sms dengan format cuaca maka aplikasi akan mengirimkan balasan sms informasi cuaca ke pengguna, jika pengguna mengirimkan sms informasi gempa bumi maka aplikasi akan mengirimkan balasan sms informasi gempa bumi ke pengguna.

Admin melakukan entri data phonebook, lihat inbox dan lihat outbox ke aplikasi informasi cuaca dan gempa bumi.

Kebutuhan sistem yang dibutuhkan dalam aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* meliputi:

1. Kebutuhan Fungsional

Aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* dapat digunakan untuk :

- a. Menerima pesan informasi cuaca dan gempa yang berasal dari inbox.
- b. Membalas pesan informasi cuaca dan gempa bumi yang dikirimkan dan meyimpan di outbox.
- c. Untuk menyimpan nomor ponsel yang di kirim informasi gempa terbaru secara otomatis di dalam phonebook .

2. Kebutuhan Data

Kebutuhan data yang digunakan dalam Aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* antara lain:

- a. Inbox
- b. SMS Cuaca
- c. SMS Gempa
- d. Outbox
- e. Phonebook
- f. URL <http://www.bmkgjateng.com/runningtext/ews.xml>

g. URL <http://www.bmkgjateng.com/runningtext/prakicu.xml>

3. Kebutuhan User

Kebutuhan user yang menggunakan Aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* antara lain:

a. Admin

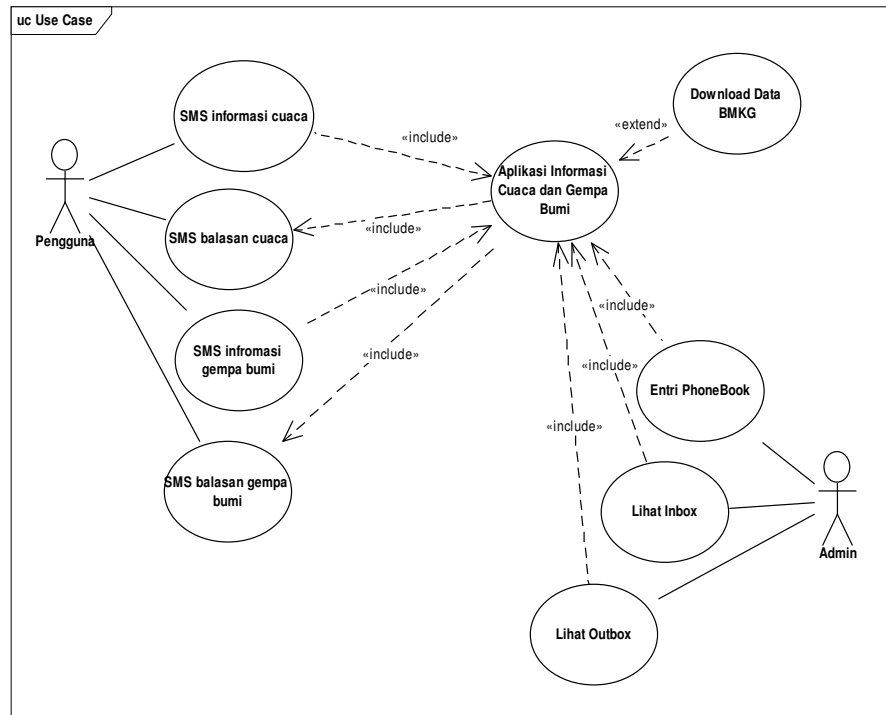
Admin bertugas untuk memasukkan data phonebook dan mengecek inbox dan outbox.

b. Pengguna

Pengguna dapat mengirimkan sms dengan format cuaca kota dan gempa ke server BMKG

4.3.2. Use Case Diagram

Use case diagram seperti gambar 4.2 terdiri dari aktor yaitu pengguna dan admin. Pengguna mengirimkan sms informasi cuaca ke aplikasi informasi cuaca dan gempa bumi dan aplikasi akan mengirimkan balasan sms informasi cuaca ke pengguna, pengguna mengirimkan sms informasi gempa bumi ke aplikasi informasi cuaca dan gempa bumi dan aplikasi akan mengirimkan balasan sms informasi gempa bumi ke pengguna. Admin melakukan entri data phonebook, lihat inbox dan lihat outbox ke aplikasi informasi cuaca dan gempa bumi.

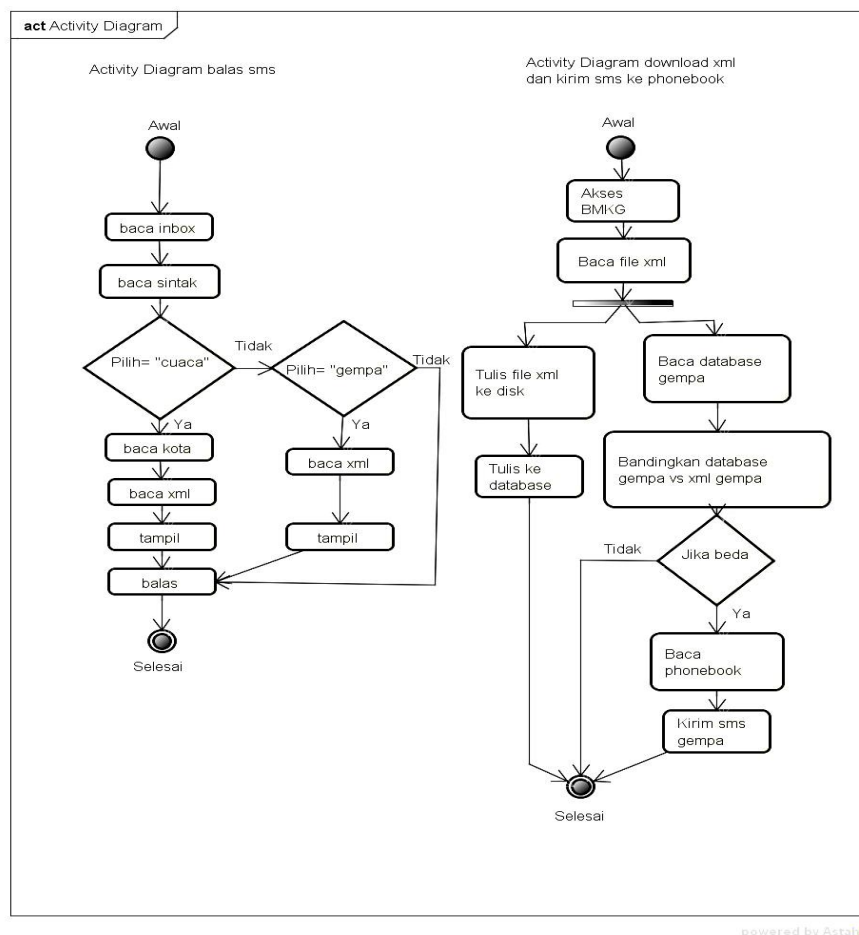


Gambar 4.2. Use Case Diagram

4.3.3. Activity Diagram

Activity diagram seperti gambar 4.3 memiliki dua activity, yaitu activity diagram balas sms dan activity diagram download xml dan kirim sms ke phonebook. Activity diagram balas sms dimulai dengan kegiatan sistem membaca inbox, jika sms berupa informasi cuaca ke aplikasi informasi cuaca dan gempa bumi maka aplikasi akan mengirimkan balasan sms informasi cuaca ke pengguna, jika sms berupa informasi gempa bumi ke aplikasi informasi cuaca dan gempa bumi atau terjadi gempa bumi maka aplikasi akan mengirimkan balasan sms informasi gempa bumi ke pengguna. Dan jika sms tidak sesuai format maka aplikasi akan mengirimkan balasan bahwa sms format yang dikirim salah.

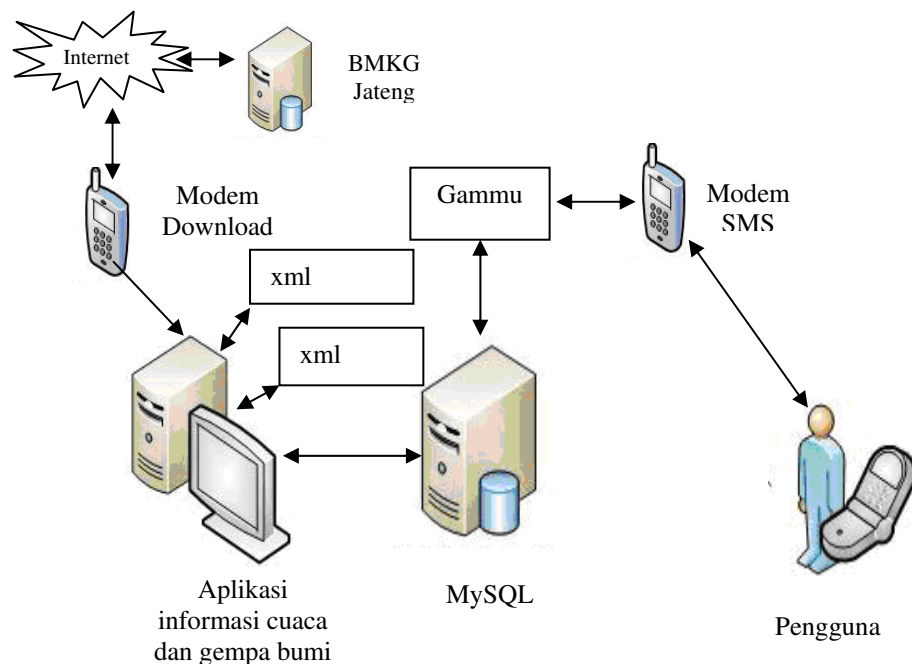
Activity diagram download xml dan kirim sms ke phonebook dimulai dengan kegiatan sistem mengakses websites BMKG lalu mendownload file XML kemudian disimpan ke disk dan disimpan database atau MYSQL. Kegiatan download tersebut dilakukan secara berulang. Pada saat download file XML dari website BMKG jika informasi gempa di file XML gempa dan di database gempa berbeda maka sistem akan mengirimkan informasi gempa ke phonebook.



Gambar 4.3. Activity Diagram

4.3.4. Arsitektur Sistem

Arsitektur sistem aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dimulai dengan pengguna mengirimkan sms dengan menggunakan handphone ke server BMKG kemudian sms diterima oleh server kemudian server akan membalas sms tersebut sesuai dengan format yang diminta kepada pengirim sms. Aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway menggunakan engine Gammu dengan melakukan konfigurasi file smsdrc dan gammurc yang terdapat pada file gammu.



Gambar 4.5. Arsitektur Sistem

Konfigurasi file gammurc yaitu mengatur port dan koneksi yang dipakai oleh modem yaitu :

device = com4:

connection = at115200

Konfigurasi file smsdrc yaitu mengatur port dan koneksi
mysql yang dipakai oleh modem yaitu :

device = com4:

connection = at115200

Service=MYSQL

user = root

password =

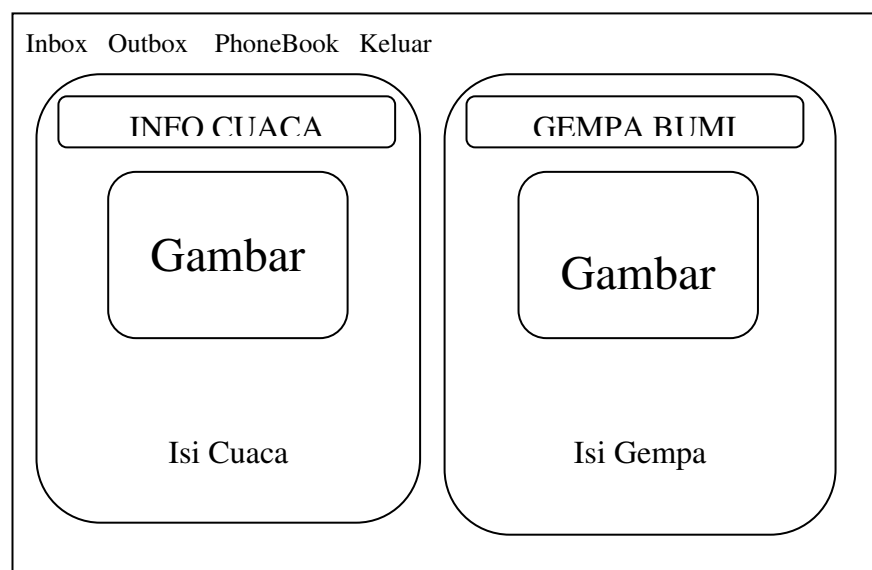
pc = localhost

database = dbpurwanto

4.3.5. Desain Input Output

1. Menu Utama

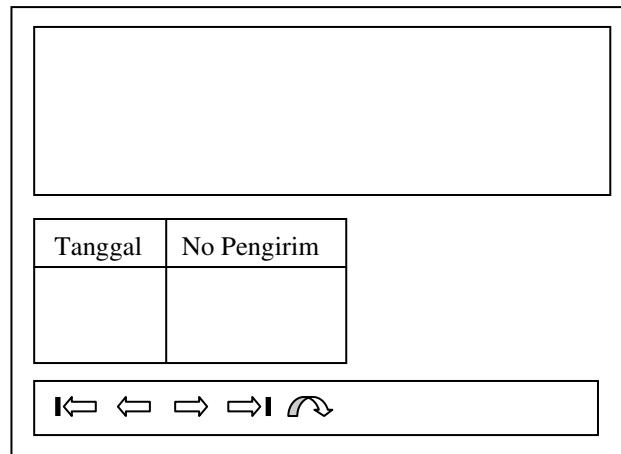
Pada menu utama seperti pada gambar 4.5 akan
ditampilkan tentang informasi cuaca dan gempa bumi yang
ditampilkan dalam bentuk animasi gif.



Gambar 4.5. Menu Utama

2. Menu Inbox

Pada menu inbox seperti pada gambar 4.6 akan ditampilkan tentang sms yang masuk ke aplikasi informasi cuaca dan gempa bumi.

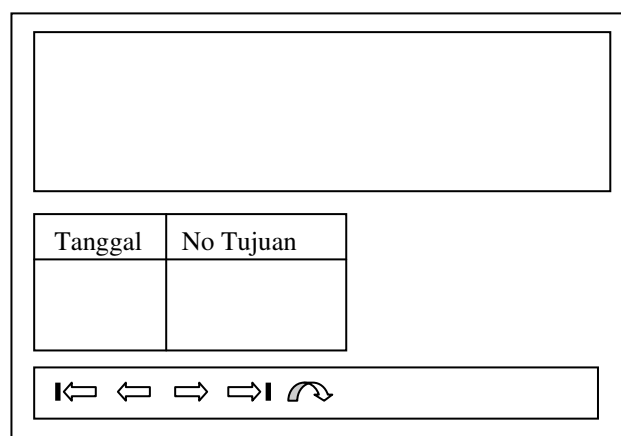


Tanggal	No Pengirim

Gambar 4.6. Menu Inbox.

3. Menu Outbox

Pada menu outbox seperti pada gambar 4.7 akan ditampilkan tentang sms yang terkirim dari aplikasi informasi cuaca dan gempa bumi ke pengirim sms.

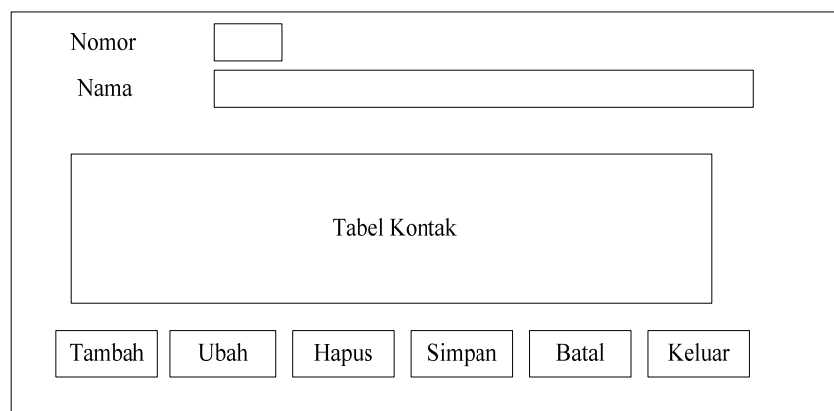


Tanggal	No Tujuan

Gambar 4.7. Menu Outbox.

4. PhoneBook

Perancangan menu phonebook seperti pada gambar 4.8 digunakan untuk memasukkan kontak yang digunakan untuk mengirimkan sms gempa. Pada halaman ini, admin dapat melakukan penambahan, perubahan, penghapusan data phonebook



The image shows a web interface for a PhoneBook. At the top, there are two input fields: one labeled 'Nomor' and another labeled 'Nama'. Below these fields is a large rectangular box labeled 'Tabel Kontak'. At the bottom of the interface, there is a horizontal row of six buttons: 'Tambah', 'Ubah', 'Hapus', 'Simpan', 'Batal', and 'Keluar'.

Gambar 4.8. Menu PhoneBook

4.3.6. Format SMS

Format sms yang dapat dikirimkan pada aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway yaitu

1. CUACA kota

Digunakan untuk mengetahui informasi cuaca di kota yang dikirimkan.

2. GEMPA

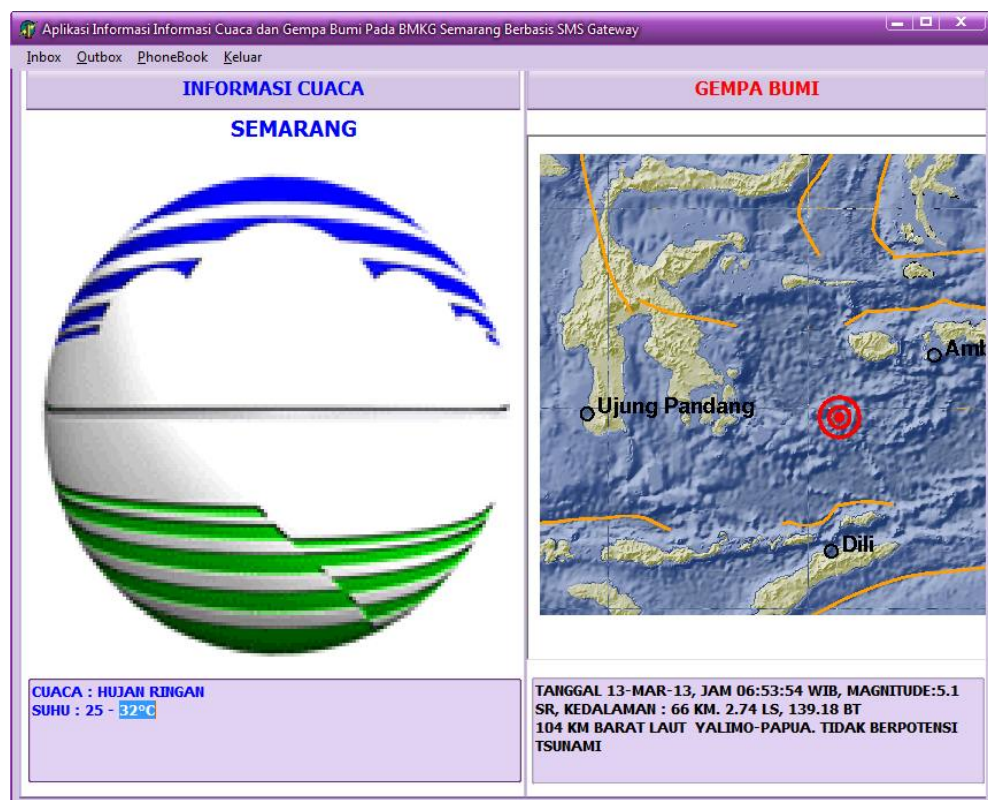
Digunakan untuk mengetahui informasi gempa bumi terkini.

BAB V

IMPLEMENTASI SISTEM

5.1. Menu Utama

Menu utama seperti pada gambar 5.1 yang merupakan implementasi dari perancangan program pada gambar 4.5 pada aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway berisi menu-menu yang digunakan untuk menghubungkan dengan form yang lain yaitu inbox yang digunakan untuk melihat data sms yang masuk, outbox yang digunakan untuk melihat sms yang keluar, dan keluar yang digunakan untuk keluar dari program



Gambar 5.1. Menu Utama

.Pada form utama akan ditampilkan informasi cuaca dan gempa bumi dimana data cuaca diambil secara otomatis oleh program dari alamat website bmkng yaitu <http://www.bmkngjateng.com/runningtext/prakicu.xml> dan hasil download file akan disimpan di direktori c:\prakicu.xml. Listing program untuk mengambil data secara otomatis yaitu

```

procedure Tfmutama.btnprakicuClick(Sender: TObject);
const
ADPNEWHOTURL='http://www.bmkngjateng.com/runningtext/prakicu.xml';
var
TmpFileName='c:\prakicu.xml';
x,z:integer ;
a,b,kota,isi,cuaca,j,h:string;
begin
if NOT Download_HTM(ADPNEWHOTURL,TmpFileName) then
begin
ShowMessage('Tidak Dapat Mendownload Prakicu.xml') ;
Exit;
end
else
begin
qycari.SQL.Clear;
qycari.SQL.Add('DELETE FROM cuaca');
qycari.ExecSQL;
XMLprakicu.FileName:='c:\prakicu.xml';
XMLprakicu.Active:=true;
tvprakicu.Items.Clear;
DomToTreeprakicu (XMLprakicu.DocumentElement, nil);
tvprakicu.FullExpand;
x:=15;
while (x<=tvprakicu.Items.Count-4) do
Begin
a:=tvprakicu.Items.Item[x].Text;
kota:=UpperCase (StringReplace(a, 'title'
', '' , [rfReplaceAll, rfIgnoreCase]));
b:=tvprakicu.Items.Item[x+1].Text;
isi:=UpperCase (StringReplace (b, 'description'
', '' , [rfReplaceAll, rfIgnoreCase]));
isi:=isi+', ' ;
h:=''; j:='';
mmcuaaca.Lines.Clear;
for z:=1 to Length(isi) do
begin
h:=copy (isi, z, 1);
if h=', ' then
begin
mmcuaaca.Lines.Add(j);
h:='';
j:='';

```

```

        Next;
    end
    else
        j:=j+h;
    end;
    tbcuaca.Append;
    tbcuacaKota.Value:=kota;
    tbcuacaCuaca.Value:=mmcuaaca.Lines.Strings[0];
    tbcuacaIsi.Value:=TrimLeft (mmcuaaca.Lines.Strings[1]);
    tbcuaca.Post;
    x:=x+5;
end;end;end;

```

Data gempa juga diambil secara otomatis oleh program dari alamat website bmkg yaitu <http://www.bmkgjateng.com/runningtext/ews.xml> dan hasil download file akan disimpan di direktori c:\gempa.xml. Listing program untuk mengambil data gempa secara otomatis yaitu

```

procedure Tfmutama.btngetempaClick(Sender: TObject);
const
    ADPNEWHOTURL='http://www.bmkgjateng.com/runningtext/ews.xml';
    TmpFileName='c:\gempa.xml';
var
    x:integer ;
    b,isi:string;
begin
    if NOT Download_HTM(ADPNEWHOTURL,TmpFileName) then
        begin
            ShowMessage('Tidak Dapat Mendownload Prakicu.xml') ;
            Exit;
        end
    else
        begin
            XMLperingatan.FileName:='c:\gempa.xml';
            XMLperingatan.Active:=true;
            tvperingatan.Items.Clear;
            DomToTreeperingatan (XMLperingatan.DocumentElement, nil);
            tvperingatan.FullExpand;
            x:=26;
            mmgempa.Lines.Clear;
            b:=tvperingatan.Items.Item[x].Text;
            isi:=UpperCase(StringReplace(b, 'description' =
            ', ', [rfReplaceAll,rfIgnoreCase]));
            mmgempa.Lines.Add(isi);

            x:=31;
            b:=tvperingatan.Items.Item[x].Text;
            isi:=UpperCase(StringReplace(b, 'description' =
            ', ', [rfReplaceAll,rfIgnoreCase]));
            mmgempa.Lines.Add(isi);
        end;
    end;
end;

```

Aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway juga akan melakukan refresh data secara otomatis setiap 300 detik dengan menggunakan komponen timer sehingga data cuaca dan gempa akan selalu terupdate dan terkini. Listing program yang digunakan untuk merefresh data yaitu

```
procedure Tfmutama.tmrrefreshTimer(Sender: TObject);
begin
    inc(i);
    if i>=300 then
    begin
        btnprakicuClick(sender);
        btngempaClick(sender);
        qygempa.Close;
        qygempa.Open;
        if mmgempa.Text<>qygempagempa.Value then
        begin
            qycari.SQL.Clear;
            qycari.SQL.Add('UPDATE gempa set gempa=:a');
            qycari.Parameters[0].Value:=mmgempa.Text;
            qycari.ExecSQL;
            qygempa.Close;
            qygempa.Open;
            Kirim1Click(sender);
        end;
        i:=0;
    end;
end;
```

Aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway menggunakan listing program untuk membalas sms yang masuk sebagai berikut :

```
procedure Tfmutama.Timer2Timer(Sender: TObject);
var
    a,b,c,d,e,f,g,h:string;
    pisah:TStringList;
    pj:integer;
begin
    qyinbox.SQL.Clear;
    qyinbox.SQL.Add('SELECT      *      FROM      INBOX      WHERE
processed="false"');
    qyinbox.Open;
    if qyinbox.RecordCount>0 then
    begin
        try
```

```

        pisah:=TStringList.Create;           // PEMISAH TEKS
        pisah.Delimiter:=' ';
        pisah.DelimitedText:=UpperCase(qyinboxTextDecoded.Value)
    ;
        a:=pisah[0];
    finally
        pisah.Free;
    end;
    if a='CUACA' then
        begin

            try
                pisah:=TStringList.Create;           // PEMISAH TEKS
                pisah.Delimiter:=' ';

                pisah.DelimitedText:=UpperCase(qyinboxTextDecoded.Value)
            ;

                b:=pisah[1];

                finally
                    pisah.Free;
                end;
                qysimpan.SQL.Clear;
                qysimpan.SQL.Add('SELECT      *      FROM      cuaca      WHERE
kota=:a');
                qysimpan.Parameters[0].Value:=b;
                qysimpan.Open;
                if qysimpan.RecordCount>0 then
                    begin
                        MemoText.Lines.Clear;
                        MemoText.Lines.Add('KOTA : '+qysimpan['kota']);
                        MemoText.Lines.Add(UpperCase(qysimpan['cuaca']));
                        MemoText.Lines.Add(UpperCase(qysimpan['isi']));
                    end
                else
                    begin
                        MemoText.Lines.Clear;
                        MemoText.Lines.Add('TIDAK ADA CUACA '+b);
                    end;
                qysimpan.SQL.Clear;
                qysimpan.SQL.Add('INSERT          INTO          outbox
(DestinationNumber, TextDecoded) VALUES (:a,:b)');

                qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
                qysimpan.Parameters[1].Value:=MemoText.Text;
                qysimpan.ExecSQL;

                qysimpan.SQL.Clear;
                qysimpan.SQL.Add('UPDATE  inbox  SET  Processed  =
"true" WHERE ID =:a');
                qysimpan.Parameters[0].Value:=qyinboxID.Value;

```

```

        qysimpan.ExecSQL;
    end
    else if a='GEMPA' then
        begin
            MemoText.Lines.Clear;
            qysimpan.SQL.Clear;
            qysimpan.SQL.Add('INSERT          INTO          outbox
(DestinationNumber, TextDecoded) VALUES (:a,:b)');

qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
qysimpan.Parameters[1].Value:=mmgempa.Text;
qysimpan.ExecSQL;






            qysimpan.SQL.Clear;
            qysimpan.SQL.Add('UPDATE  inbox  SET  Processed  =
"true" WHERE ID =:a');
            qysimpan.Parameters[0].Value:=qyinboxID.Value;
            qysimpan.ExecSQL;

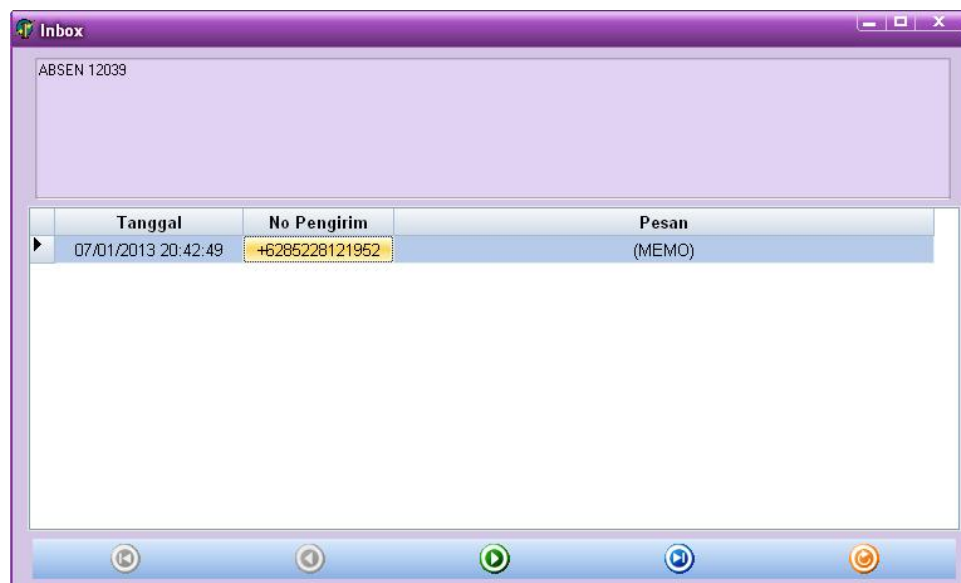
        end
    else
        begin
            MemoText.Lines.Clear;
            MemoText.Lines.Add('Keyword  Yang  Anda  Masukkan
Salah');
            qysimpan.SQL.Clear;
            qysimpan.SQL.Add('INSERT          INTO          outbox
(DestinationNumber, TextDecoded) VALUES (:a,:b)');
qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
qysimpan.Parameters[1].Value:=MemoText.Text;
qysimpan.ExecSQL;
            qysimpan.SQL.Clear;
            qysimpan.SQL.Add('UPDATE  inbox  SET  Processed  =
"true" WHERE ID =:a');
            qysimpan.Parameters[0].Value:=qyinboxID.Value;
            qysimpan.ExecSQL;
        end;
    end;
end;

```

5.2. Menu Inbox






Menu inbox seperti gambar 5.5 yang merupakan implementasi dari perancangan program pada gambar 4.6 digunakan untuk melihat sms yang masuk ke aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway. Pada form inbox terdapat 5 tombol yaitu

tombol  untuk melihat data awal sms yang masuk, tombol  untuk melihat data sms masuk sebelumnya, tombol  untuk melihat data sms masuk selanjutnya, tombol  untuk melihat data terakhir sms masuk dan tombol  untuk merefresh tabel inbox



Gambar 5.2. Menu Inbox

5.3. Menu Outbox

Menu outbox seperti gambar 5.6 yang merupakan implementasi dari perancangan program pada gambar 4.7 digunakan untuk melihat sms yang keluar dari aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway. Pada form outbox terdapat 5 tombol yaitu tombol  untuk melihat data awal sms yang keluar, tombol  untuk melihat data sms keluar sebelumnya, tombol  untuk melihat data sms keluar selanjutnya, tombol  untuk melihat data terakhir sms keluar dan tombol  untuk merefresh tabel outbox



Gambar 5.3. Menu Outbox

5.4. Menu PhoneBook

Form phonebook seperti gambar 5.7 yang merupakan implementasi dari perancangan program pada gambar 4.8 digunakan untuk memasukkan data kontak yang digunakan untuk mengirimkan sms gempal. Pada form phonebook terdapat 6 tombol yang dapat digunakan yaitu tombol tambah untuk menambah data phonebook, tombol ubah untuk mengubah data phonebook, tombol hapus untuk menghapus data phonebook, tombol keluar untuk keluar dari form phonebook, tombol simpan untuk menyimpan data phonebook, tombol batal untuk membatalkan pengisian data phonebook

The screenshot shows a Java Swing window titled "Kontak" with a purple title bar. Inside the window, there are two text input fields: "Nomor" with the value "+6285226227246" and "Nama" with the value "Purwanto". Below these fields is a table with two columns: "Nomor HP" and "Nama". The table contains one row with the values "+6285226227246" and "Purwanto". At the bottom of the window is a toolbar with six buttons: "Tambah" (with a plus icon), "Ubah" (with a pencil icon), "Hapus" (with a trash icon), "Simpan" (with a floppy disk icon), "Batal" (with a red X icon), and "Keluar" (with a door icon).

Nomor HP	Nama
+6285226227246	Purwanto

Gambar 5.4. Menu PhoneBook

5.5. Kebutuhan Sistem

Setelah dilakukan penelitian terhadap aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway*, maka kebutuhan sistem meliputi kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*).

5.5.1. Kebutuhan Perangkat Keras

Aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway*, diperlukan sebuah unit komputer yang digunakan untuk mengelola proses data cuaca dan gempa bumi, serta untuk aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway* yang akan dibuat. Adapun spesifikasi standar minimal perangkat keras yang akan digunakan sebagai berikut :

- a. Processor Intel (R) Pentium dual core
- b. Memory 1 GB – 6300 DD3
- c. HDD 250 GB – 7500 ATA
- d. Monitor 15”
- e. Keyboard dan Mouse
- f. UPS
- g. Modem

5.5.2. Kebutuhan Perangkat Lunak

Untuk dapat beroperasi secara optimal, komputer tersebut harus memenuhi spesifikasi yang diperlukan serta diperlukan

dukungan dari perangkat lunak (*software*) sesuai dengan kebutuhannya agar program aplikasi dapat berjalan dengan baik.

Kebutuhan perangkat lunak tersebut antara lain :

- a. Microsoft Windows XP sebagai sistem operasinya.
- b. Borland Delphi untuk membuat aplikasi informasi cuaca dan gempa bumi dengan menggunakan *SMS Gateway*.
- c. Gammu yang digunakan untuk menghubungkan modem dengan MySQL

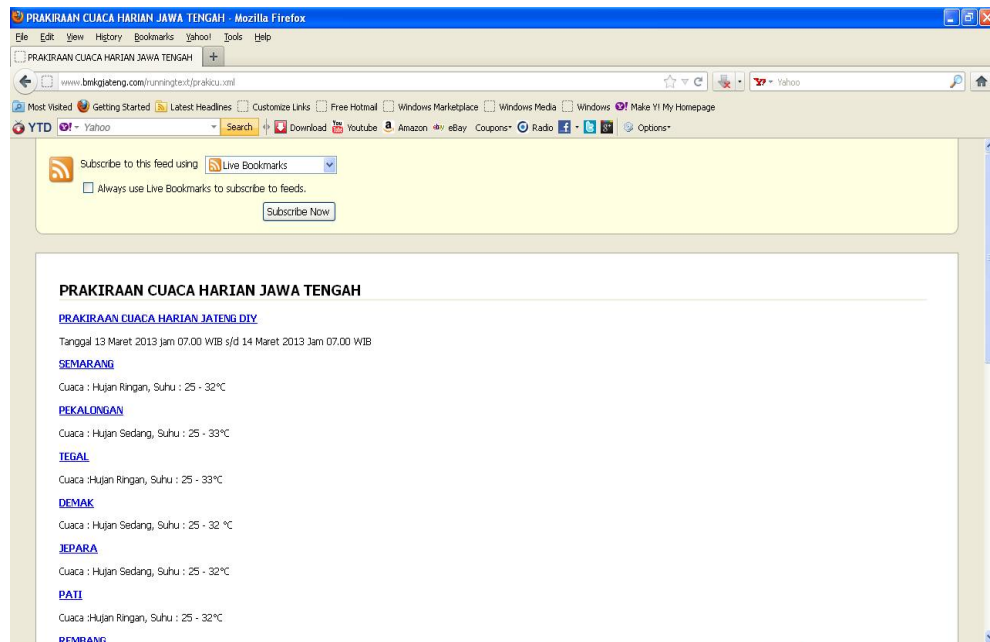
BAB VI

HASIL PENELITIAN DAN PEMBAHASAN

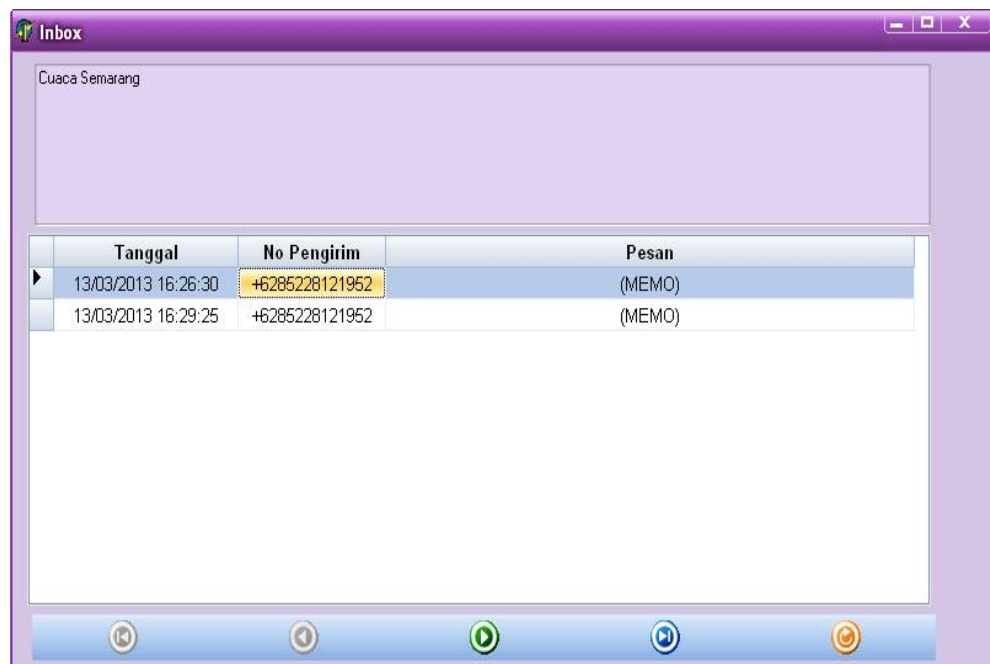
6.1. Hasil Penelitian

Aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dapat digunakan untuk membalas sms yang masuk secara otomatis sesuai dengan format yang telah ditentukan. Sistem ini membalas sms sesuai informasi cuaca dan gempa bumi diambil langsung dari website BMKG secara otomatis sehingga data cuaca dan gempa selalu uptodate dan terkini. Pengiriman sms gempa dilakukan oleh sistem secara otomatis ke nomor hp yang diinput di phonebook jika website www.bmkgjateng.com mengupdate informasi gempa terbaru.

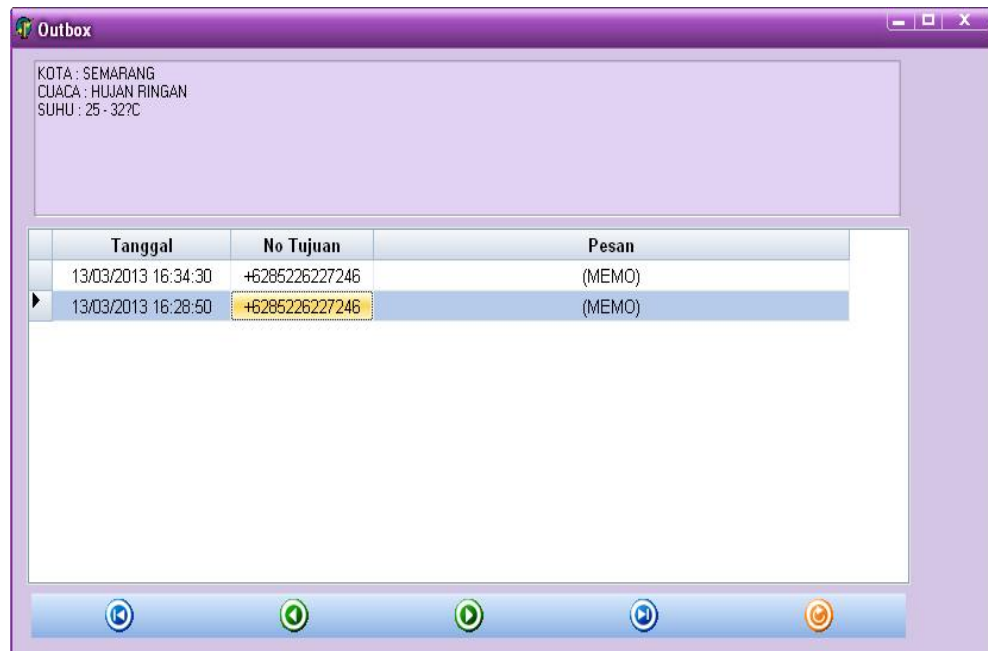
Aplikasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway mempunyai dua format sms yang dapat diterima yaitu CUACA SPASI KOTA dan GEMPA. Jika terdapat sms masuk dengan format CUACA SPASI KOTA maka server akan membalas sms tersebut dengan kondisi cuaca yang dikirimkan seperti pada gambar 6.4.



Gambar 6.1. Informasi Cuaca di Website BMKG



Gambar 6.2. Inbox Permintaan Informasi Cuaca

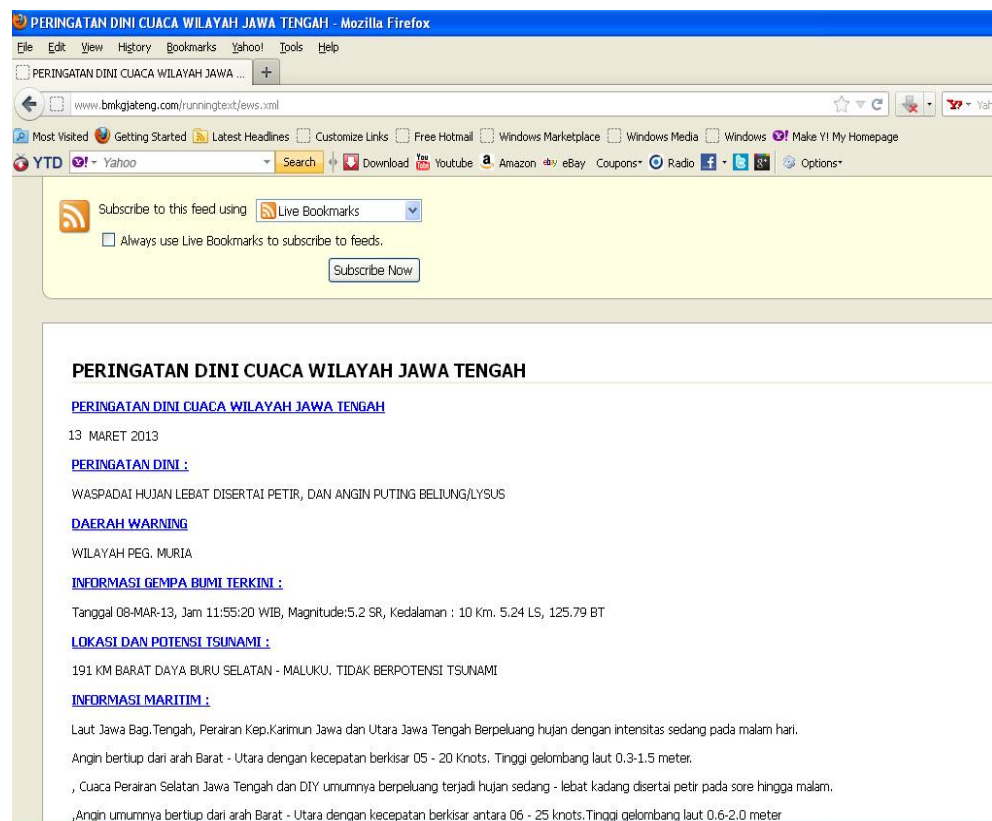


Gambar 6.3. Outbox Balasan Cuaca

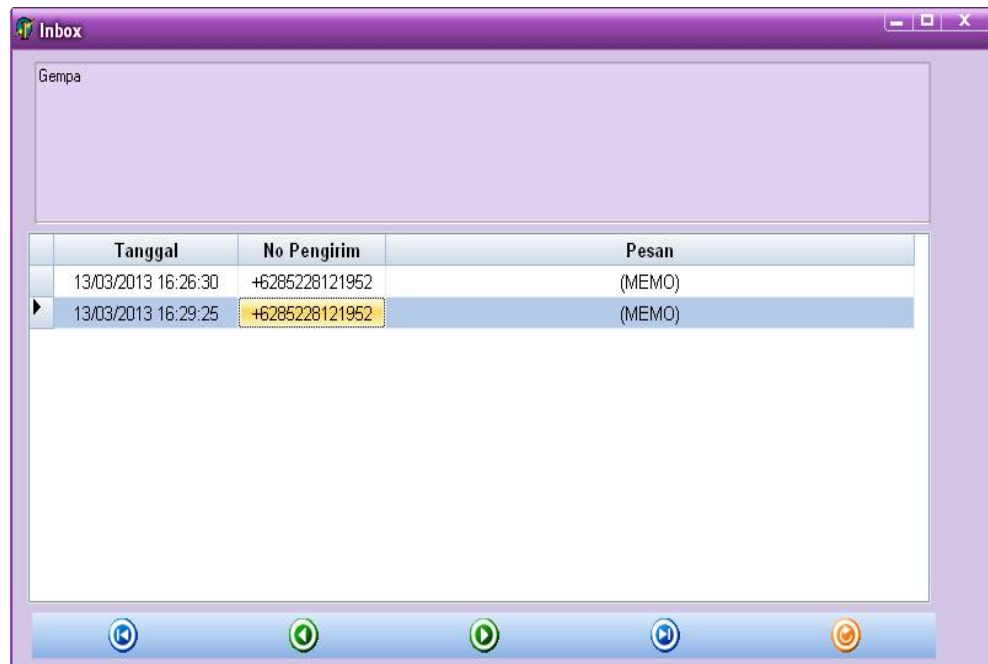


Gambar 6.4. SMS Balasan Cuaca

Jika terdapat sms masuk dengan format GEMPA maka server akan membalas sms tersebut dengan informasi gempa bumi terakhir yang terjadi seperti pada gambar 6.8.



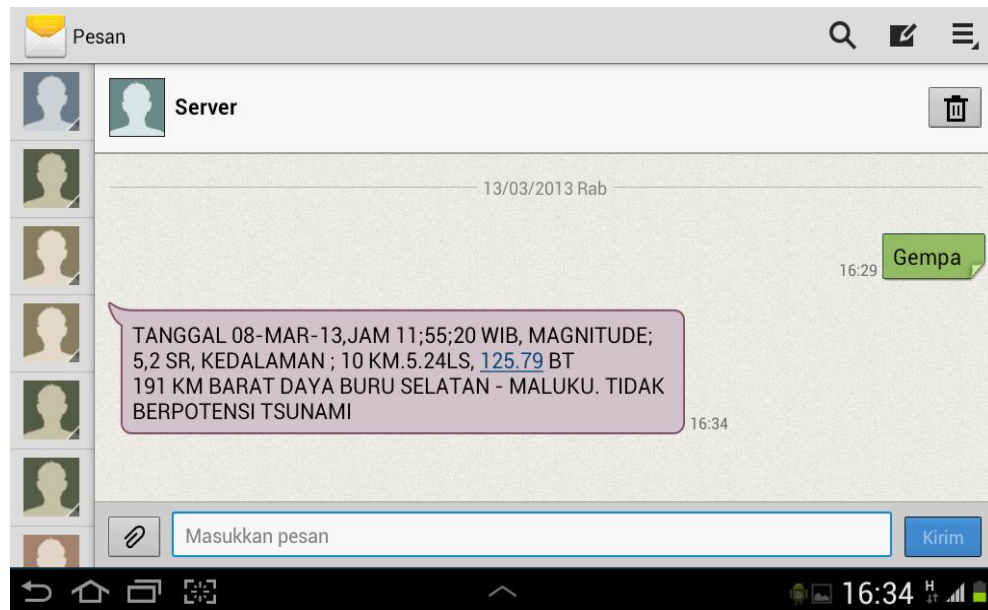
Gambar 6.5. Informasi Gempa Pada Website BMKG



Gambar 6.6. Inbox Permintaan Informasi Gempa

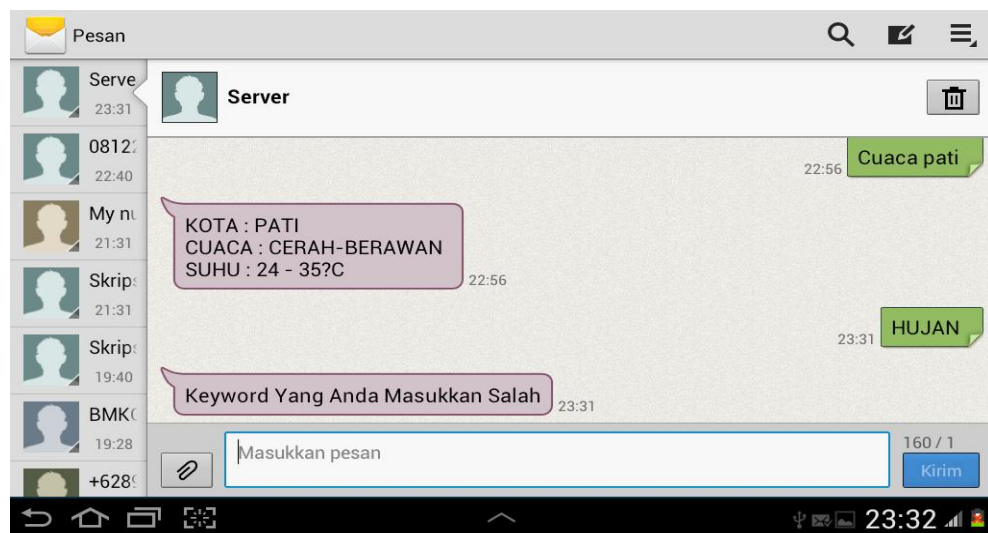


Gambar 6.7. Outbox Balasan Gempa



Gambar 6.8. SMS Balasan Gempa

Selain itu server akan membalas dengan sms 'Keyword Yang Anda Masukkan Salah' seperti pada gambar 6.9.

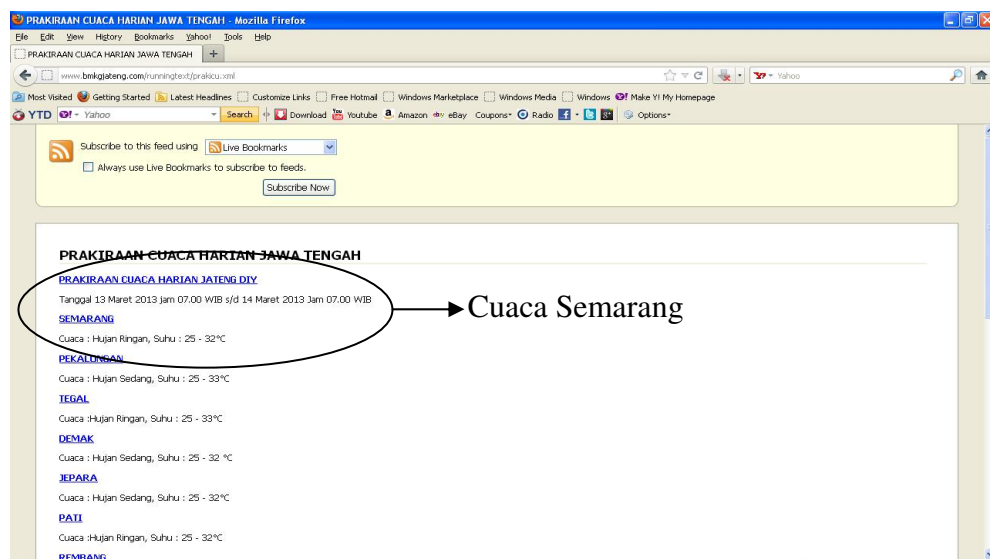


Gambar 6.9. SMS Format Salah

6.2. Pembahasan

Dari hasil penelitian diatas dapat disimpulkan bahwa informasi cuaca dan gempa bumi yang ditampilkan dari BMKG Semarang sama dengan informasi SMS cuaca dan gempa bumi yang dikirimkan ke pengguna dari tanggal yang dikirimkan.

1. Cuaca

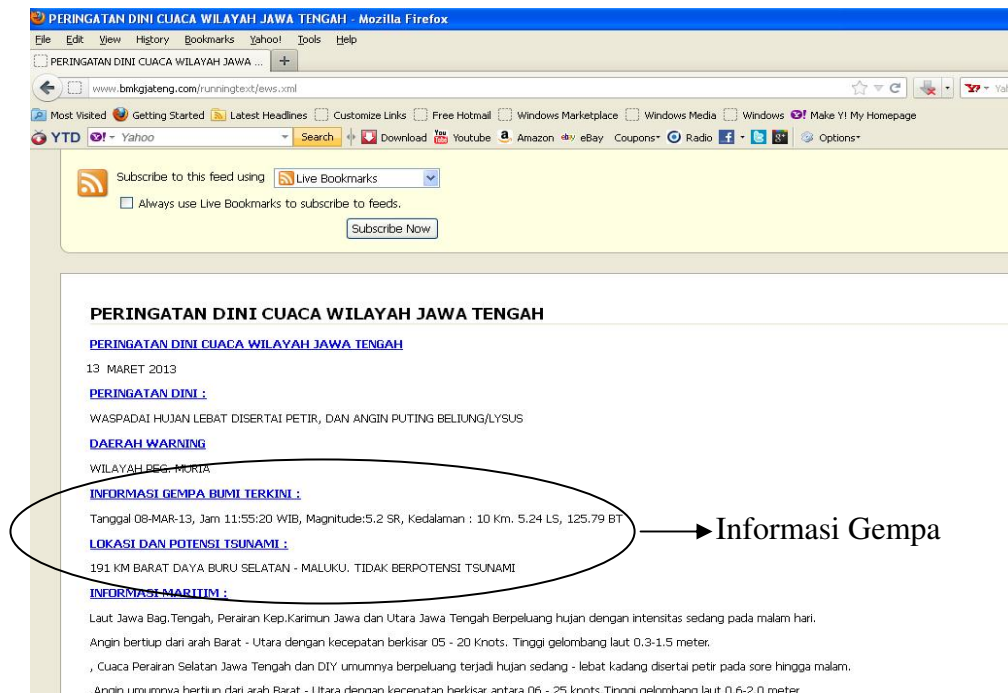


Gambar 6.10. Informasi Cuaca Pada Website BMKG

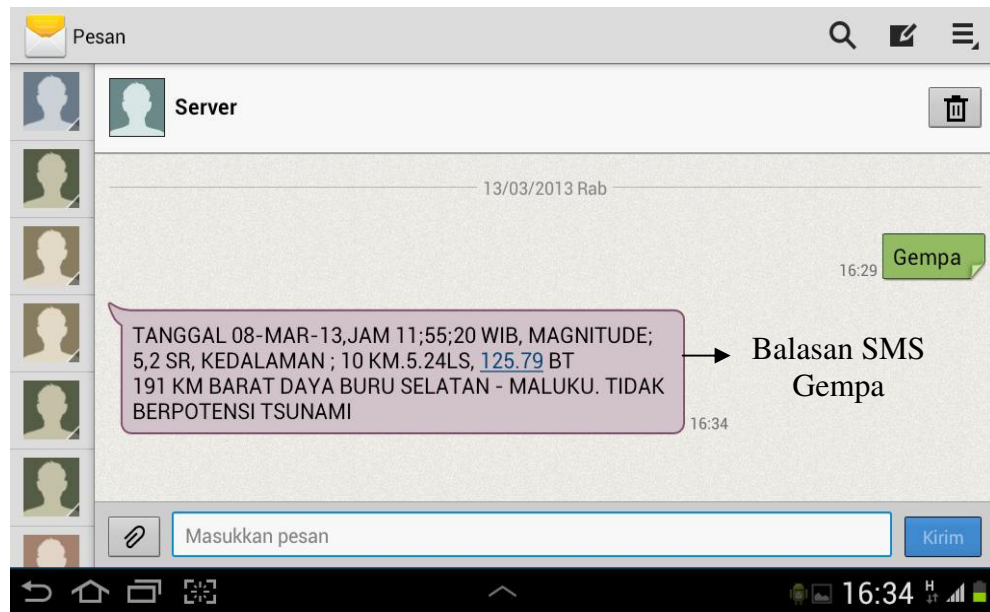


Gambar 6.11. SMS Cuaca dan XML Cuaca BMKG

2. Gempa



Gambar 6.12. Informasi Gempa Pada website BMKG



Gambar 6.13. SMS Gempa dan XML Gempa BMKG

BAB VII

PENUTUP

Berdasarkan hasil implementasi yang telah dilakukan dalam pembuatan aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway, dapat ditarik beberapa kesimpulan dan saran dari penelitian ini yaitu.

6.1. Kesimpulan

1. Aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dapat digunakan untuk membalas sms yang masuk secara otomatis sesuai dengan format yang telah ditentukan.
2. Kelebihan dari informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway yaitu sistem ini dapat langsung secara otomatis membalas sms ketika sms yang dikirimkan sesuai dengan format yang telah ditentukan dan informasi cuaca dan gempa bumi diambil langsung dari website BMKG secara otomatis sehingga data cuaca dan gempa selalu uptodate dan terkini.
3. Aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dibuat dengan menggunakan Delphi dan XML dengan menggunakan engine GAMMU untuk menghubungkan modem dengan database MySQL.

6.2. Saran

Beberapa saran yang dapat digunakan untuk pengembangan sistem selanjutnya adalah

1. Aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway dapat dikembangkan lagi dengan menggunakan teknologi seperti wap.
2. Aplikasi informasi informasi cuaca dan gempa bumi pada BMKG Semarang berbasis SMS Gateway sebaiknya menggunakan nomor handphone yang mudah diingat atau dengan menggunakan 4 digit angka diantaranya dengan bekerjasama dengan provider-provider layanan sms.
3. Data-data yang sudah lama sebaiknya dibackup guna untuk menghindari kehilangan data bila terjadi kerusakan pada sistem atau pada perangkat keras.

DAFTAR PUSTAKA

- BMKG, 2012, *Pengertian Cuaca*, www.bmkg.go.id, diunduh pada tanggal 12/11/2012
- BMKG, 2012, *Gempa Bumi*, www.bmkg.go.id, diunduh pada tanggal 12/11/2012
- Budiarso, Ir. Zuly, M.Cs, Eddy Nurraharjo, S.T., M.Cs, 2012, *Sistem Monitoring Tingkat Ketinggian Air Bendungan Berbasis Mikrokontroller*, Unisbank Semarang
- Budicahyanto, Dwi, 2003, *SMS Gateway*, PT. Elex Media Komputindo, Jakarta
- Gammu, 2012, *Gammu*, <http://wammu.eu/gammu/>, diunduh pada tanggal 12/11/2012
- Gunadi, 2002, *Visual Modelling Menggunakan UML dan Rational Rose*, Penerbit Informatika, Bandung
- Pranata, Antony, 2002, *Dasar Pemrograman Delphi 6.0*, Andi Offset, Yogyakarta.
- Prasetyo, Didik Dwi, 2003, *Administrasi Database Server MySQL*, Elex Media Komputindo, Jakarta.
- Rosely, Ely, 2011, *Aplikasi Informasi Cuaca Dan Alert Bencana Alam Berbasis Sms Gateway*, Politeknik Telkom Bandung
- Tejakusuma, Iwan G., 2008, *Analisis Pasca Bencana Tsunami Ciamis-Cilacap*, Pusat Teknologi Sumberdaya Lahan, Wilayah Dan Mitigasi Bencana Deputi Bidang TPSA - BPPT
- Whitten, Jeffery L., 2004, *Metode Desain dan Analisa Sistem*, Andi Offset, Yogyakarta

LAMPIRAN I

LISTING PROGRAM

```
unit Utama;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, sSkinManager, xmldom, XMLIntf, Menus, msxmldom,
  XMLDoc, SHDocVw, ExtActns,
  ComCtrls, StdCtrls, ADODB, DB, OleCtrls, ExtCtrls, sMemo,
  sLabel,
  GIFImage, sPanel;

type
  Tfmutama = class(TForm)
    sSkinManager1: TsSkinManager;
    XMLprakicu: TXMLDocument;
    XMLperingatan: TXMLDocument;
    MainMenu1: TMainMenu;
    Inbox1: TMenuItem;
    Outbox1: TMenuItem;
    Keluar1: TMenuItem;
    tvprakicu: TTreeView;
    tvperingatan: TTreeView;
    btnprakicu: TButton;
    ADOConnection1: TADOConnection;
    qycari: TADOQuery;
    tbcuaca: TADOTable;
    mmcuaaca: TMemo;
    tbcuacakota: TStringField;
    tbcuacacuaca: TStringField;
    tbcuacaisi: TMemoField;
    tmrrefresh: TTimer;
    sPanel2: TsPanel;
    Image2: TImage;
    sLabelFX4: TsLabelFX;
    mmcuaacal: TsMemo;
    sPanel5: TsPanel;
    sPanel3: TsPanel;
    mmgempa: TsMemo;
    sPanel6: TsPanel;
    WebBrowser1: TWebBrowser;
    qycuaca: TADOQuery;
    qycuacakota: TStringField;
    qycuacacuaca: TStringField;
    qycuacaisi: TMemoField;
    Timer1: TTimer;
    Timer2: TTimer;
```

```

qyinbox: TADOQuery;
qyinboxUpdatedInDB: TDateTimeField;
qyinboxReceivingDateTime: TDateTimeField;
qyinboxText: TMemoField;
qyinboxSenderNumber: TStringField;
qyinboxCoding: TStringField;
qyinboxUDH: TMemoField;
qyinboxSMSCNumber: TStringField;
qyinboxClass: TIntegerField;
qyinboxTextDecoded: TMemoField;
qyinboxID: TAutoIncField;
qyinboxRecipientID: TMemoField;
qyinboxProcessed: TStringField;
qysimpan: TADOQuery;
MemoText: TMemo;
PhoneBook1: TMenuItem;
PopupMenu1: TPopupMenu;
Kirim1: TMenuItem;
qy kirim: TADOQuery;
btngempa: TButton;
mmg: TsMemo;
qygempa: TADOQuery;
qygempagempa: TMemoField;
procedure Keluar1Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure btngempaClick(Sender: TObject);
procedure btnprakicuClick(Sender: TObject);
procedure tmrrefreshTimer(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Inbox1Click(Sender: TObject);
procedure Outbox1Click(Sender: TObject);
procedure PhoneBook1Click(Sender: TObject);
procedure Kirim1Click(Sender: TObject);
private
    { Private declarations }
    procedure DomToTreeprakicu(XmlNode: IXMLNode; TreeNode:
TTreeNode);
    procedure DomToTreeperingatan(XmlNode: IXMLNode; TreeNode:
TTreeNode);
public
    { Public declarations }
end;

var
    fmutama: Tfmutama;
    z,i:integer;
implementation
uses XmlIntfDefinition, Inbox, Outbox, Kontak;
{$R *.dfm}

function Download_HTM(const sURL, sLocalFileName : string):
boolean;
begin
    Result := True;
    with TDownloadURL.Create(nil) do

```

```

try
    URL:=sURL;
    Filename:=sLocalFileName;
    try
        ExecuteTarget(nil) ;
    except
        Result:=False
    end;
finally
    Free;
end;
end;

procedure Tfmutama.DomToTreeprakicu (XmlNode: IXMLNode; TreeNode:
TTreeNode);
var
    I: Integer;
    NewTreeNode: TTreeNode;
    NodeText: string;
    AttrNode: IXMLNode;
begin
    // skip text nodes and other special cases
    if XmlNode.NodeType <> ntElement then
        Exit;
    // add the node itself
    NodeText := XmlNode.NodeName;
    if XmlNode.IsTextElement then
        NodeText := NodeText + ' = ' + XmlNode.NodeValue;
    NewTreeNode := tvprakicu.Items.AddChild(TreeNode, NodeText);
    // add attributes
    for I := 0 to XmlNode.AttributeNodes.Count - 1 do
        begin
            AttrNode := XmlNode.AttributeNodes.Nodes[I];
            tvprakicu.Items.AddChild(NewTreeNode,
                '[' + AttrNode.NodeName + ' = ' + AttrNode.Text + '"]');
        end;
    // add each child node
    if XmlNode.HasChildNodes then
        for I := 0 to XmlNode.ChildNodes.Count - 1 do
            DomToTreeprakicu (XmlNode.ChildNodes.Nodes [I],
NewTreeNode);
        end;

procedure Tfmutama.DomToTreeperingatan(XmlNode: IXMLNode;
TreeNode: TTreeNode);
var
    I: Integer;
    NewTreeNode: TTreeNode;
    NodeText: string;
    AttrNode: IXMLNode;
begin
    if XmlNode.NodeType <> ntElement then
        Exit;
    NodeText := XmlNode.NodeName;
    if XmlNode.IsTextElement then
        NodeText := NodeText + ' = ' + XmlNode.NodeValue;

```

```

NewTreeNode := tvperingatan.Items.AddChild(TreeNode, NodeText);
for I := 0 to xmlNode.AttributeNodes.Count - 1 do
begin
    AttrNode := xmlNode.AttributeNodes.Nodes[I];
    tvperingatan.Items.AddChild(NewTreeNode,
        '[' + AttrNode.NodeName + ' = ' + AttrNode.Text + ']' );
end;
// add each child node
if XmlNode.HasChildNodes then
    for I := 0 to xmlNode.ChildNodes.Count - 1 do
        DomToTreeperingatan (xmlNode.ChildNodes.Nodes [I],
NewTreeNode);
end;

```

```

procedure Tfmutama.Keluar1Click(Sender: TObject);
begin
close;
end;

```

```

procedure Tfmutama.FormActivate(Sender: TObject);
begin
i:=0;z:=0;
    tbcuaca.Close;
    tbcuaca.Open;
    btngempaClick(sender);
qygempa.Close;
qygempa.Open;
//mmg.Text:=qygempagempa.Value;
if mmgempa.Text<>qygempagempa.Value then
    begin
        qycari.SQL.Clear;
        qycari.SQL.Add('UPDATE gempa set gempa=:a');
        qycari.Parameters[0].Value:=mmgempa.Text;
        qycari.ExecSQL;

        Kirim1Click(sender);
    end;

```

```

    btnprakicuClick(sender);
    WebBrowser1.Navigate('http://data.bmkg.go.id/eqmap.gif');
    qygempa.Close;
    qygempa.Open;
    mmg.Text:=qygempagempa.Value;

```

```

    z:=0;
    qycuaca.SQL.Clear;
    qycuaca.SQL.Add('SELECT * FROM cuaca');
    qycuaca.Open;
    qycuaca.First;
    mmcuacal.Text:=qycuacacuaca.Value;
    mmcuacal.Lines.Add(qycuacaisi.Value);
    sLabelFX4.Caption:=qycuacakota.Value;

```

```

    if mmcuacal.Lines.Strings[0]=Trim('CUACA : BERAWAN') then

```

```

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\berawan.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerah.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH-BERAWAN')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerahberawa
n.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH -
BERAWAN') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerahberawa
n.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA :HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
    else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')

    else

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\logo.gif');

end;

procedure Tfmutama.btngempaClick(Sender: TObject);
const
//download url
    ADPNEWHOTURL='http://www.bmkgjateng.com/runningtext/ews.xml';
    TmpFileName='c:\gempa.xml';
var
x:integer ;
b,isi:string;
begin
    if NOT Download_HTM(ADPNEWHOTURL,TmpFileName) then
        begin
            ShowMessage('Tidak Dapat Mendownload Prakicu.xml') ;
            Exit;
        end
    else
        begin
            XMLperingatan.FileName:='c:\gempa.xml';

```

```

XMLperingatan.Active:=true;
tvperingatan.Items.Clear;
DomToTreeperingatan (XMLperingatan.DocumentElement, nil);
tvperingatan.FullExpand;

x:=26;

mmgempa.Lines.Clear;
b:=tvperingatan.Items.Item[x].Text;
isi:=UpperCase(StringReplace(b, 'description =
', '', [rfReplaceAll, rfIgnoreCase]));
mmgempa.Lines.Add(isi);

x:=31;
b:=tvperingatan.Items.Item[x].Text;
isi:=UpperCase(StringReplace(b, 'description =
', '', [rfReplaceAll, rfIgnoreCase]));
mmgempa.Lines.Add(isi);

end;

end;

procedure Tfmutama.btnprakicuClick(Sender: TObject);
const
//download url

ADPNEWHOTURL='http://www.bmkgjateng.com/runningtext/prakicu.xml';
TmpFileName='c:\prakicu.xml';
var
x,z:integer ;
a,b,kota,isi,cuaca,j,h:string;
begin
if NOT Download_HTM(ADPNEWHOTURL,TmpFileName) then
begin
ShowMessage('Tidak Dapat Mendownload Prakicu.xml') ;
Exit;
end
else
begin
qycari.SQL.Clear;
qycari.SQL.Add('DELETE FROM cuaca');
qycari.ExecSQL;
XMLprakicu.FileName:='c:\prakicu.xml';
XMLprakicu.Active:=true;
tvprakicu.Items.Clear;
DomToTreeprakicu (XMLprakicu.DocumentElement, nil);
tvprakicu.FullExpand;
x:=15;
while (x<=tvprakicu.Items.Count-4) do
Begin
a:=tvprakicu.Items.Item[x].Text;
kota:=UpperCase(StringReplace(a, 'title =
', '', [rfReplaceAll, rfIgnoreCase]));
b:=tvprakicu.Items.Item[x+1].Text;

```

```

isi:=UpperCase(StringReplace(b,'description =
','',[rfReplaceAll,rfIgnoreCase]));
isi:=isi+', '
h:=''; j:='';
mmcuaaca.Lines.Clear;
for z:=1 to Length(isi) do
begin
h:=copy(isi,z,1);
if h=', ' then
begin
mmcuaaca.Lines.Add(j);
h:='';
j:='';
Next;
end
else
j:=j+h;
end;

tbcuaca.Append;
tbcuacaKota.Value:=kota;
tbcuacaCuaca.Value:=mmcuaaca.Lines.Strings[0];
tbcuacaIsi.Value:=TrimLeft(mmcuaaca.Lines.Strings[1]);
tbcuaca.Post;

x:=x+5;
end;
end;

```

```

end;

```

```

procedure Tfmutama.tmrrefreshTimer(Sender: TObject);
begin
inc(i);
//sPanel5.Caption:=IntToStr(i);
if i>=300 then
begin
btnprakicuClick(sender);
btngempaClick(sender);
qygempa.Close;
qygempa.Open;

if mmgempa.Text<>qygempagempa.Value then
begin
qycari.SQL.Clear;
qycari.SQL.Add('UPDATE gempa set gempa=:a');
qycari.Parameters[0].Value:=mmgempa.Text;
qycari.ExecSQL;
qygempa.Close;
qygempa.Open;
// mmg.Text:=qygempagempa.Value;
Kirim1Click(sender);
end;
i:=0;

```

```

end;

end;

procedure Tfmutama.Timer1Timer(Sender: TObject);
begin
inc(z);
if z=100 then
begin
qycuaca.Next;
if qycuaca.Eof then
qycuaca.First;

mmcuacal.Lines.Clear;
mmcuacal.Text:=qycuacacuaca.Value;
mmcuacal.Lines.Add(qycuacaisi.Value);
sLabelFX4.Caption:=qycuacakota.Value;
if mmcuacal.Lines.Strings[0]=Trim('CUACA : BERAWAN') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\berawan.gif
')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerah.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH-BERAWAN')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerahberawa
n.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : CERAH -
BERAWAN') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\cerahberawa
n.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN') then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA :HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')
else if mmcuacal.Lines.Strings[0]=Trim('CUACA : HUJAN SEDANG')
then

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\hujan.gif')

else

Image2.picture.LoadFromFile('d:\SKRIPSI\Program\Images\logo.gif');

```



```

        z:=0;
    end;

end;

procedure Tfmutama.Timer2Timer(Sender: TObject);
var
    a,b,c,d,e,f,g,h:string;
    pisah:TStringList;
    pj:integer;
begin
    qyinbox.SQL.Clear;
    qyinbox.SQL.Add('SELECT * FROM INBOX WHERE processed="false"');
    qyinbox.Open;
    if qyinbox.RecordCount>0 then
        begin
            try
                pisah:=TStringList.Create;           // PEMISAH TEKS
                pisah.Delimiter:=' ';
                pisah.DelimitedText:=UpperCase(qyinboxTextDecoded.Value);
                a:=pisah[0];
            finally
                pisah.Free;
            end;
            if a='CUACA' then
                begin
                    try
                        pisah:=TStringList.Create;           // PEMISAH TEKS
                        pisah.Delimiter:=' ';
                        pisah.DelimitedText:=UpperCase(qyinboxTextDecoded.Value);
                        b:=pisah[1];

                        finally
                            pisah.Free;
                        end;
                    qysimpan.SQL.Clear;
                    qysimpan.SQL.Add('SELECT * FROM cuaca WHERE kota=:a');
                    qysimpan.Parameters[0].Value:=b;
                    qysimpan.Open;
                    if qysimpan.RecordCount>0 then
                        begin
                            MemoText.Lines.Clear;
                            MemoText.Lines.Add('KOTA : '+qysimpan['kota']);
                            MemoText.Lines.Add(Uppercase(qysimpan['cuaca']));
                            MemoText.Lines.Add(Uppercase(qysimpan['isi']));
                        end
                    else
                        begin
                            MemoText.Lines.Clear;
                            MemoText.Lines.Add('TIDAK ADA CUACA '+b);
                        end;
                    qysimpan.SQL.Clear;
                    qysimpan.SQL.Add('INSERT INTO outbox (DestinationNumber,
TextDecoded) VALUES (:a,:b)');

```

```

        qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
        qysimpan.Parameters[1].Value:=MemoText.Text;
        qysimpan.ExecSQL;

        qysimpan.SQL.Clear;
        qysimpan.SQL.Add('UPDATE inbox SET Processed = "true" WHERE
ID =:a');
        qysimpan.Parameters[0].Value:=qyinboxID.Value;
        qysimpan.ExecSQL;
    end
    else if a='GEMPA' then
    begin
        MemoText.Lines.Clear;
        qysimpan.SQL.Clear;
        qysimpan.SQL.Add('INSERT INTO outbox (DestinationNumber,
TextDecoded) VALUES (:a,:b)');
        qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
        qysimpan.Parameters[1].Value:=mmgempa.Text;
        qysimpan.ExecSQL;

        qysimpan.SQL.Clear;
        qysimpan.SQL.Add('UPDATE inbox SET Processed = "true" WHERE
ID =:a');
        qysimpan.Parameters[0].Value:=qyinboxID.Value;
        qysimpan.ExecSQL;

    end

    else
    begin
        MemoText.Lines.Clear;
        MemoText.Lines.Add('Keyword Yang Anda Masukkan Salah');

        qysimpan.SQL.Clear;
        qysimpan.SQL.Add('INSERT INTO outbox (DestinationNumber,
TextDecoded) VALUES (:a,:b)');
        qysimpan.Parameters[0].Value:=qyinboxSenderNumber.Value;
        qysimpan.Parameters[1].Value:=MemoText.Text;
        qysimpan.ExecSQL;

        qysimpan.SQL.Clear;
        qysimpan.SQL.Add('UPDATE inbox SET Processed = "true" WHERE
ID =:a');
        qysimpan.Parameters[0].Value:=qyinboxID.Value;
        qysimpan.ExecSQL;
    end;
    end;
end;

procedure Tfmutama.Inbox1Click(Sender: TObject);
begin
    fminbox.ShowModal;
end;

procedure Tfmutama.Outbox1Click(Sender: TObject);
begin

```

```

fmoutbox.ShowDialog;
end;

procedure Tfmutama.PhoneBook1Click(Sender: TObject);
begin
fmkontak.ShowDialog;
end;

procedure Tfmutama.Kirim1Click(Sender: TObject);
begin
qy kirim.SQL.Clear;
qy kirim.SQL.Add('DELETE FROM kirim');
qy kirim.ExecSQL;

qy kirim.SQL.Clear;
qy kirim.SQL.Add('INSERT INTO kirim(nomor) SELECT nomor FROM
kontak');
qy kirim.ExecSQL;

qy kirim.SQL.Clear;
qy kirim.SQL.Add('UPDATE kirim set isi=:a');
qy kirim.Parameters[0].Value:=mmngempa.Text;
qy kirim.ExecSQL;

qy kirim.SQL.Clear;
qy kirim.SQL.Add('INSERT INTO outbox (DestinationNumber,
TextDecoded) select * from kirim');
qy kirim.ExecSQL;

end;

end.

```

```

unit Inbox;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, DB, ADOdb, ImgList, acAlphaImageList, Grids, BaseGrid,
  AdvGrid,
  DBAdvGrid, ExtCtrls, DBAdvNavigator, StdCtrls, sMemo;

type
  Tfminbox = class(TForm)
    btnnavigator: TDBAdvNavigator;
    grdinbox: TDBAdvGrid;
    dsinbox: TDataSource;
    ImageList32: TsAlphaImageList;
    tbinbox: TADOTable;
    tbinboxUpdatedInDB: TDateTimeField;
    tbinboxReceivingDateTime: TDateTimeField;
    tbinboxText: TMemoField;
    tbinboxSenderNumber: TStringField;
    tbinboxCoding: TStringField;
    tbinboxUDH: TMemoField;
    tbinboxSMSCNumber: TStringField;
    tbinboxClass: TIntegerField;
    tbinboxTextDecoded: TMemoField;
    tbinboxID: TAutoIncField;
    tbinboxRecipientID: TMemoField;
    tbinboxProcessed: TStringField;
    sMemo1: TsMemo;
    procedure FormActivate(Sender: TObject);
    procedure btnnavigatorBtnRefresh(Sender: TObject);
    procedure tbinboxTextDecodedGetText(Sender: TField; var Text:
String;
      DisplayText: Boolean);
    procedure grdinboxClickCell(Sender: TObject; ARow, ACol:
Integer);
    procedure grdinboxKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure btnnavigatorClick(Sender: TObject; Button:
TAdvNavigateBtn);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fminbox: Tfminbox;

implementation

uses Utama;

{$R *.dfm}

```

```

procedure Tfminbox.FormActivate(Sender: TObject);
begin
    tbinbox.Close;
    tbinbox.Open;
    grdinbox.Refresh;

end;

procedure Tfminbox.btnnavigatorBtnRefresh(Sender: TObject);
begin
    tbinbox.Close;
    tbinbox.Open;
    grdinbox.Refresh;
end;

procedure Tfminbox.tbinboxTextDecodedGetText(Sender: TField;
    var Text: String; DisplayText: Boolean);
begin
    Text := Copy(tbinboxTextDecoded.Value, 1, 50);
    tbinbox.Close;
    tbinbox.Open;
end;

procedure Tfminbox.grdinboxClickCell(Sender: TObject; ARow, ACol:
Integer);
begin
    sMem01.Lines.Text:=tbinboxTextDecoded.Value;
end;

procedure Tfminbox.grdinboxKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    sMem01.Lines.Text:=tbinboxTextDecoded.Value;
end;

procedure Tfminbox.btnnavigatorClick(Sender: TObject;
    Button: TAdvNavigateBtn);
begin
    sMem01.Lines.Text:=tbinboxTextDecoded.Value;
end;

end.

```

```

unit Outbox;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, DB, ADODB, ImgList, acAlphaImageList, Grids, BaseGrid,
  AdvGrid,
  DBAdvGrid, ExtCtrls, DBAdvNavigator, StdCtrls, sMemo;

type
  Tfmoutbox = class(TForm)
    btnnavigator: TDBAdvNavigator;
    grdoutbox: TDBAdvGrid;
    dsoutbox: TDataSource;
    ImageList32: TsAlphaImageList;
    tboutbox: TADOTable;
    tboutboxUpdatedInDB: TDateTimeField;
    tboutboxInsertIntoDB: TDateTimeField;
    tboutboxSendingDateTime: TDateTimeField;
    tboutboxDeliveryDateTime: TDateTimeField;
    tboutboxText: TMemoField;
    tboutboxDestinationNumber: TStringField;
    tboutboxCoding: TStringField;
    tboutboxUDH: TMemoField;
    tboutboxSMSCNumber: TStringField;
    tboutboxClass: TIntegerField;
    tboutboxTextDecoded: TMemoField;
    tboutboxID: TIntegerField;
    tboutboxSenderID: TStringField;
    tboutboxSequencePosition: TIntegerField;
    tboutboxStatus: TStringField;
    tboutboxStatusError: TIntegerField;
    tboutboxTPMR: TIntegerField;
    tboutboxRelativeValidity: TIntegerField;
    tboutboxCreatorID: TMemoField;
    sMemo1: TsMemo;
    procedure FormActivate(Sender: TObject);
    procedure tboutboxTextDecodedGetText(Sender: TField; var Text:
String;
      DisplayText: Boolean);
    procedure grdoutboxClickCell(Sender: TObject; ARow, ACol:
Integer);
    procedure btnnavigatorClick(Sender: TObject; Button:
TAdvNavigateBtn);
    procedure grdoutboxKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  fmoutbox: Tfmoutbox;

```

implementation

```
{ $R *.dfm }
```

```
procedure Tfmoutbox.FormActivate(Sender: TObject);
begin
  tboutbox.Close;
  tboutbox.Open;
  grdoutbox.Refresh;
end;

procedure Tfmoutbox.tboutboxTextDecodedGetText(Sender: TField;
  var Text: String; DisplayText: Boolean);
begin
  Text := Copy(tboutboxTextDecoded.AsString, 1, 50);
  tboutbox.Close;
  tboutbox.Open;

end;

procedure Tfmoutbox.grdoutboxClickCell(Sender: TObject; ARow,
  ACol: Integer);
begin
  sMem01.Lines.Text:=tboutboxTextDecoded.Value;
end;

procedure Tfmoutbox.btnnavigatorClick(Sender: TObject;
  Button: TAdvNavigateBtn);
begin
  sMem01.Lines.Text:=tboutboxTextDecoded.Value;
end;

procedure Tfmoutbox.grdoutboxKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  sMem01.Lines.Text:=tboutboxTextDecoded.Value;
end;

end.
```

```

unit Kontak;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, Buttons, sBitBtn, ComCtrls,
  sComboBox,
  sEdit, Grids, BaseGrid, AdvGrid, DBAdvGrid, jpeg, ExtCtrls,
  ImgList,
  ToolWin;

type
  Tfmkontak = class(TForm)
    Label3: TLabel;
    Label4: TLabel;
    grdkontak: TDBAdvGrid;
    edtnomor: TsEdit;
    edtnama: TsEdit;
    qycari: TADOQuery;
    qykontak: TADOQuery;
    dslokasi: TDataSource;
    ToolBar1: TToolBar;
    btntambah: TToolButton;
    btnubah: TToolButton;
    btnhapus: TToolButton;
    btnkeluar: TToolButton;
    ToolButton1: TToolButton;
    btnsimpan: TToolButton;
    btnbatal: TToolButton;
    ToolButton2: TToolButton;
    ImageList1: TImageList;
    qykontaknomor: TStringField;
    qykontaknama: TStringField;
    procedure FormActivate(Sender: TObject);
    procedure btnkeluarClick(Sender: TObject);
    procedure btntambahClick(Sender: TObject);
    procedure btnsimpanClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnubahClick(Sender: TObject);
    procedure btnhapusClick(Sender: TObject);
    procedure btnbatalClick(Sender: TObject);
    procedure grdkontakClickCell(Sender: TObject; ARow, ACol:
Integer);
    procedure grdkontakKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure edtnomorKeyPress(Sender: TObject; var Key: Char);
    procedure edtalamatKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var

```



```

    fmkontak: Tfmkontak;
    ket:string;
implementation

uses Utama;

{$R *.dfm}

procedure Tfmkontak.FormActivate(Sender: TObject);
begin
    qykontak.Close;
    qykontak.Open;
    edtnomor.Enabled := false;
    edtnama.Enabled := false;

    btnsimpan.Enabled := false;
    btnbatal.Enabled := false;
    btntambah.Enabled := true;
    btnkeluar.Enabled := true;
    ket := '';
    FormCreate(sender);
    if qykontak.IsEmpty=true then
    begin
        btnhapus.Enabled:=false;
        btnubah.Enabled := false;
        grdkontak.Enabled:=false;
        grdkontak.Refresh;
    end
    else
    begin
        btnhapus.Enabled:=true;
        btnubah.Enabled := true;
        grdkontak.Enabled:=true;
        grdkontak.Refresh;
    end;
end;

procedure Tfmkontak.btnkeluarClick(Sender: TObject);
begin
    qykontak.Close;
    close;
end;

procedure Tfmkontak.btntambahClick(Sender: TObject);
var
    a:string;
begin
    ket := 'ISI';
    btnbatal.Enabled:=true;
    btntambah.Enabled:=false;
    btnkeluar.Enabled:=false;
    btnhapus.Enabled:=false;
    btnubah.Enabled:=false;
    grdkontak.Enabled:=false;
    edtnomor.Enabled := True;

```

```

edtnama.Enabled := True;

edtnomor.SetFocus;
btnsimpan.Enabled:=true;
qykontak.Last;
formcreate(sender);

end;

procedure Tfmkontak.btnsimpanClick(Sender: TObject);
begin
if ket='ISI' then
begin
qycari.SQL.Clear;
qycari.SQL.Add('INSERT INTO kontak VALUES (:a, :b)');
qycari.Parameters[0].Value:=edtnomor.Text;
qycari.Parameters[1].Value:=edtnama.Text;
qycari.ExecSQL;
qykontak.Close;
qykontak.Open;
grdkontak.Refresh;
btntambahClick(sender);
end
else if ket='EDIT' then
begin
qycari.SQL.Clear;
qycari.SQL.Add('UPDATE kontak SET nama=:a WHERE nomor=:j');
qycari.Parameters[1].Value:=edtnomor.Text;
qycari.Parameters[0].Value:=edtnama.Text;
qycari.ExecSQL;
FormActivate(sender);
end;
end;

procedure Tfmkontak.FormCreate(Sender: TObject);
begin
edtnomor.text:= '';
edtnama.text:= '';
end;

procedure Tfmkontak.btnubahClick(Sender: TObject);
begin
if (qykontak.IsEmpty=true) or (edtnomor.Text='') then
begin
ShowMessage('Data Tidak Dapat Diubah'+#13+'Nomor Masih Kosong');
FormActivate(sender);
end
else
begin
ket:='EDIT';
btnsimpan.Enabled:=true;
btnbatal.Enabled:=true;
btntambah.Enabled:=false;
btnkeluar.Enabled:=false;
btnhapus.Enabled:=false;
btnubah.Enabled:=false;
end;
end;

```

```

        grdkontak.Enabled:=false;
        edtnama.Enabled := True;
        edtnama.SetFocus;
    end;

end;

procedure Tfmkontak.btnhapusClick(Sender: TObject);
begin
    if (qykontak.IsEmpty=true) or (edtnomor.Text='') then
        begin
            ShowMessage('Data Tidak Dapat Dihapus'+#13+'Nomor Masih Kosong');
            FormActivate(sender);
        end
    else
        begin
            if MessageDlg('Data Ingin Dihapus', mtwarning, [mbYes, mbNo],
0) = mrYes then
                begin
                    qycari.SQL.Clear;
                    qycari.SQL.Add('DELETE FROM kontak WHERE nomor=:a');
                    qycari.Parameters[0].Value:=edtnomor.Text;
                    qycari.ExecSQL;
                    FormActivate(sender);
                end;
            end;
        end;
end;

procedure Tfmkontak.btnbatalClick(Sender: TObject);
begin
    FormActivate(sender);
end;

procedure Tfmkontak.grdkontakClickCell(Sender: TObject; ARow,
    ACol: Integer);
begin
    edtnomor.Text:=Trim(qykontaknomor.Value);
    edtnama.Text:=Trim(qykontaknama.Value);
end;

procedure Tfmkontak.grdkontakKeyDown(Sender: TObject; var Key:
    Word;
    Shift: TShiftState);
begin
    edtnomor.Text:=Trim(qykontaknomor.Value);
    edtnama.Text:=Trim(qykontaknama.Value);

end;

procedure Tfmkontak.edtnomorKeyPress(Sender: TObject; var Key:
    Char);
begin
    if key=chr(13) then
        edtnama.SetFocus;
    end;
end;

```

```
procedure Tfmkontak.edtalamatKeyPress(Sender: TObject; var Key:
Char);
begin
    if key=chr(13) then
        if MessageDlg('Data Sudah Benar', mtInformation, [mbYes, mbNo],
0) = mrYes then
            btnsimpanClick(sender)
        else
            edtnama.SetFocus;
    end;
end.
```