

BAB II

LANDASAN TEORI

1.7 Perkembangan Teknologi Telepon Seluler

Seiring dengan berjalannya waktu dan meningkatnya kebutuhan komunikasi bergerak, maka tingkat kepemilikan telepon seluler (ponsel) juga meningkat cukup tinggi. Hal ini dapat dilihat dari tingkat penggunaan fasilitas SMS pada hari-hari besar nasional misalnya pada hari raya Idul Fitri dan Natal yang mencapai 50 sampai 70 juta SMS dalam sehari.

Besarnya pasar pengguna ponsel telah mendorong sejumlah produsen peralatan telekomunikasi seperti Nokia, Sony Ericsson, Motorola, Samsung dan lain-lain untuk berlomba memperebutkan peluang tersebut dengan memproduksi ponsel dengan beraneka bentuk dan inovasi teknologinya. Secara umum, inovasi ponsel dilakukan dengan mengembangkan protokol, koneksi dan aplikasinya.

1.8 Mengenal Short Message Service

Short Message Service (SMS) merupakan layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (nir kabel), memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antar terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal seperti e-mail, paging, voice mail dan lain-lain. Isu SMS pertama kali muncul di belahan Eropa pada tahun 1991 bersama sebuah teknologi komunikasi *wireless* yang saat ini cukup banyak penggunaannya, yaitu *Global System for Mobile Communication*

(GSM). Dipercaya bahwa pesan pertama yang dikirim menggunakan SMS dilakukan pada bulan Desember 1992, dikirim dari sebuah *Personal Computer* (PC) ke telepon *mobile* dalam jaringan GSM milik Vodafone Inggris. Perkembangan kemudian merambah ke benua Amerika, dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti Bell Sputh Mobility, PrimeCo, Nextel, dan beberapa operator lain. Teknologi digital yang digunakan sangat bervariasi dari yang berbasis GSM, *Time Division Multiple Access* (TDMA), hingga *Code Division Multiple Access* (CDMA).

Tidak diragukan lagi SMS sangat sukses dipasaran, ditempat kelahirannya sendiri, yaitu Eropa trafik SMS mencapai lebih dari 3 miliar *message* per bulan meskipun tidak ada program marketing yang proaktif dari operator seluler dan vendor pembuat perangkat komunikasi bergerak. Kesuksesan SMS dianggap sebagai kesuksesan yang tidak disengaja dan cukup mengejutkan bagi pihak-pihak yang terjun dalam industri telekomunikasi bergerak, karena beberapa pihak yang berkompeten sebelumnya memprediksikan bahwa SMS tidak akan berlaku karena penggunaannya cukup sulit dan materi marketingnya sulit ditemukan.

SMS menjadi fenomena tersendiri, dalam waktu yang cukup singkat tingkat pertumbuhannya sangat tinggi tanpa ada penurunan tarif yang berarti. Biasanya dalam kasus layanan telepon bergerak, tarif akan turun seiring dengan meningkatnya pengguna. Fakta lainnya adalah fasilitas SMS dalam telepon bergerak ternyata mempunyai andil yang cukup besar dalam menarik kaum muda masuk ke pasar telepon bergerak.

1.9 Cara Kerja SMS

Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah melakukan pengiriman short message dari satu terminal pelanggan ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Centre (SMSC)*, disebut juga *Message Centre (MC)*. SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward* trafik short message. Didalamnya termasuk penentuan atau pencarian rute tujuan akhir dari sort message.



Gambar 2.1 Skema cara kerja SMS

Layanan SMS merupakan sebuah layanan yang bersifat *non-real time* dimana sebuah short message dapat di *submit* ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu short message hingga sampai ke tujuan.

Kegagalan pengiriman yang bersifat sementara seperti tujuan yang tidak diaktifkan selalu teridentifikasi sehingga pengiriman ulang *short message* akan selalu dilakukan kecuali bila diberlakukan aturan bahwa *short message* yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

Karakteristik utama dari SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket dengan *bandwidth* kecil. Dengan karakteristik ini, pengiriman suatu data yang pendek dapat dilakukan dengan efisiensi yang sangat tinggi. Pada awalnya SMS diciptakan untuk menggantikan layanan paging dengan menyediakan layanan serupa yang bersifat *two-way messaging* ditambah dengan *notification service*, khususnya untuk *voice mail*. Pada perkembangan selanjutnya, muncul jenis-jenis layanan lain seperti mail, fax, dan paging integration, interactive banking, information service, dan integrasi dengan aplikasi berbasis internet. Integrasi dengan aplikasi berbasis internet mendorong timbulnya layanan seperti web-based messaging, gaming dan chatting.

Pada era kompetensi global saat ini, perbedaan layanan merupakan faktor yang cukup signifikan untuk mencapai sukses service provider. Sekali sebuah layanan tergelar seperti telepon, maka SMS merupakan sebuah senjata yang cukup ampuh dalam rangka diferensiasi layanan. Bahkan jika pasar menerima dengan antusias, maka tidak mustahil SMS akan menjadi sumber pendapatan baru bagi service provider atau operator telekomunikasi.

2.4 SMS-Gateway

Berbeda dengan aplikasi SMS Broadcast yang bersifat satu arah, SMS Gateway adalah jenis layanan dua arah, artinya selain dapat menerima pesan dari luar juga dapat mengirim balasan secara otomatis ke nomor tujuan, contohnya SMS quiz, SMS polling, dan lain-lain (Wahidin, 2010). SMS Gateway dapat menangani berbagai proses bisnis dan bidang layanan, diantaranya sebagai berikut:

- 1.9.1.1.1. Manajemen Inventori
2. CRM (*Customer Relationship Management*), misalnya Rumah Sakit, Rumah Makan, Café, Executive Club, Stasiun Radio, Stasiun TV, Lembaga Pendidikan, dan lain-lain
3. Call Center dan SMS Pengaduan, misalnya Polisi, PLN, Instansi Pemerintah.

SMS Gateway memanfaatkan arsitektur teknologi komunikasi SMS untuk menerapkan aplikasi bernilai tambah dengan memanfaatkan komunikasi SMS untuk optimalisasi proses bisnis perusahaan dan peningkatan kualitas layanan dari institusi pelayanan publik. Beberapa kemampuan SMS Gateway, yaitu untuk :

1. Memperbesar skala aplikasi teknologi informasi dengan menggunakan komunikasi SMS interaktif
2. Menyediakan aplikasi kolaborasi komunikasi SMS berbasis web untuk pengguna di institusi atau perusahaan

3. Menjangkau konsumen maupun pengguna jasa layanan institusi atau perusahaan secara mudah menggunakan komunikasi SMS interaktif.

2.5 Konsep Dasar Sistem dan Informasi

Penerapan sistem informasi pada sebuah perusahaan, lembaga, instansi pemerintah, universitas, organisasi, maupun suatu kegiatan wirausaha merupakan salah satu bentuk dari adanya kegiatan yang berjalan melalui prosedur-prosedur tertentu dan sudah menjadi ketentuan umum dalam perusahaan atau lembaga tersebut untuk dicapainya suatu informasi yang berguna bagi pelaksana manajemen.

2.5.1 Pengertian Sistem

Sistem didefinisikan berbeda-beda menurut beberapa ahli. *Murdick dan Ross (1993)* dalam *Hanif Al Fatta (2007)* mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Dalam kamus *Webster's Unbridged* sistem adalah elemen-elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi.

Menurut *Scott (1996)*, sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Beberapa definisi sistem tersebut di atas, secara umum memiliki kesamaan di mana sistem terdiri dari

beberapa bagian yang bekerja sama untuk mencapai satu tujuan yang disepakati bersama.

2.5.2 Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya:

1. **Batasan (*Boundary*)** : Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. **Lingkungan (*environment*)** : Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. **Masukan (*input*)** : Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. **Keluaran (*output*)** : Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. **Komponen (*component*)** : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. **Penghubung (*interface*)** : Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

7. **Penyimpanan** (*storage*) : Area yang dikuasi dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.5.3 Pengertian Informasi

Informasi sangat penting artinya bagi suatu sistem yang akan dibuat dalam sebuah organisasi. Suatu organisasi akan terus tumbuh jika informasi-informasi yang diperoleh berkualitas karena pengambilan keputusan-keputusan penting yang menyangkut masa depan organisasi selalu berdasarkan informasi yang ada.

McLeod (1995) dalam *Hanif Al Fatta (2007)* mengatakan bahwa informasi adalah data yang telah diproses, atau data yang memiliki arti. Sumber dari informasi adalah data. Data dalam hal ini adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

2.5.4 Pengertian Sistem Informasi

Salah satu unsur penting dalam pengambilan keputusan dalam suatu organisasi adalah informasi. Sebelum mengambil keputusan perlu diketahui darimana dan bagaimana informasi ini sendiri diperoleh. Pada dasarnya informasi tersebut dapat diperoleh dari sistem informasi.

Menurut *Abdul Kadir (2003)*, sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. Sedangkan menurut *Hanif Al Fatta (2007)*, sistem informasi merupakan sistem dengan komponen-komponen yang bekerja untuk mengolah data menjadi informasi.

Berdasarkan teori-teori di atas, maka dapat disimpulkan bahwa sistem informasi adalah komponen-komponen yang terkait dalam sistem dan saling berhubungan untuk mengubah data menjadi informasi guna mencapai sasaran atau tujuan.





2.6 Perancangan Sistem dan Database

Perancangan sistem adalah diagram yang menggambarkan sistem yang sedang berjalan dan sistem baru yang akan digunakan dengan menggunakan komputer. Dalam tahap-tahap ini dilakukan pemecahan masalah secara logika dengan menggunakan alat bantu, yaitu DFD, ERD, dan Normalisasi.

2.6.2 DFD (Data Flow Diagram)

Data Flow Diagram adalah gambaran sistem secara logika. Gambaran ini tidak tergantung perangkat keras, perangkat lunak, struktur data atau organisasi file. Keuntungan dari data flow yaitu, memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan. Ada 4 simbol data flow diagram, yaitu :

Tabel 2.1 Simbol DFD

Simbol DFD	Keterangan
	SOURCE (KESATUAN LUAR)
	PROSES
	DATA FLOW (ARUS DATA)
	DATA STORE (SIMPANAN DATA)

Proses pada data flow diagram dapat merupakan sekumpulan program, satu modul atau sub program, dapat juga merupakan file elemen dari satu database atau satu bagian record, penyimpanan data dapat juga berupa disk *magnetic drum* dan *magnetic tape*.

2.6.3 Pengertian Database

Database berfungsi layaknya kabinet tempat menyimpan arsip-arsip. Perbedaanya adalah bahwa dalam kabinet, data berbentuk lembaran kertas (*hard copy*), sedangkan dalam database, data berbentuk elektronik yang tersimpan dalam komputer, tepatnya di media penyimpanan bernama hard disk.

Database adalah kumpulan file-file atau tabel-tabel yang saling berelasi atau berhubungan antara satu dengan yang lain. Relasi tersebut ditunjukkan adanya kunci dari tiap file atau tabel yang ada. Dalam satu database biasanya terdiri dari kumpulan data tiap pada suatu kantor atau perusahaan yang saling berhubungan. (Andi Sunyoto, 2007).

2.6.4 Pengertian Flowchart

Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. Flowchart diawali dengan penerimaan input, pemrosesan input dan diakhiri dengan penampilan output. Simbol-simbol flowchart ada beberapa macam, yaitu :

Tabel 2.2 Simbol-simbol flowchart

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Pemulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROCESS	Proses perhitungan/proses pengolahan data

	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Pemulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

2.6.5 Operasi-operasi Dasar dalam Database

Operasi-operasi dasar yang dapat dilakukan berkenaan dengan database adalah sebagai berikut :

- a. Pembuatan database baru (*create database*)
- b. Penghapusan database (*drop database*)
- c. Pembuatan file/tabel dari suatu database (*create table*)
- d. Penghapusan file/tabel dari suatu database (*drop table*)
- e. Penambahan atau pengisian data baru ke dalam sebuah file/tabel di sebuah database (*insert*)
- f. Perubahan data dari sebuah file/tabel (*update*)
- g. Penghapusan data data dari sebuah file/tabel (*delete*)

2.6.6 Teknik Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal (Kusrini, 2007).

Dalam perspektif normalisasi sebuah database dikatakan baik jika setiap tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal, jika :

- a. Jika ada dekomposisi/penguraian tabel, maka dekomposisinya dijamin aman (*lossless-join decomposition*)
- b. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*dependency preservation*)
- c. Tidak melanggar *Boyce Code Normal Form* (BCNF), jika tidak bisa minimal tidak melanggar bentuk normalisasi ketiga

Alasan utama dari normalisasi database minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya "*insertion anomalies*", "*deletion anomalies*", dan "*update anomalies*". Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada database yang tidak normal.

"*Insertion anomalies*" adalah sebuah kesalahan dalam penempatan informasi entry data baru ke seluruh tempat dalam database dimana informasi

tersebut perlu disimpan. Dalam database yang telah dinormalisasi, proses pemasukan suatu informasi baru hanya perlu dimasukkan ke dalam satu tempat.

"*Deletion anomalies*" adalah sebuah kesalahan dalam penghapusan suatu informasi dalam database harus dilakukan dengan penghapusan informasi tersebut dari beberapa tempat dalam database. Dalam database yang telah dinormalisasi, penghapusan suatu informasi hanya perlu dilakukan dalam satu tempat dalam database tersebut.

Sedangkan dalam melakukan *update* satu informasi, kesalahan juga dapat terjadi ketika kita harus melakukan *update* ke seluruh tempat yang menyimpan informasi tersebut. Kesalahan ini disebut dengan "*update anomalies*".

Berikut adalah bentuk-bentuk normalisasi :

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

b. Bentuk normal tahap pertama (1st Normal Form)

Bentuk normal pertama memiliki ciri yaitu setiap data dibentuk dalam flat file (file datar/rata), data bentuk, dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang atribut bernilai ganda.

c. Bentuk normal tahap kedua (2nd Normal Form)

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key).

d. Bentuk normal tahap ketiga (*3rd Normal Form*)

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsionalitas dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah superkey pada tabel tersebut
- Atau A merupakan bagian dari primary key pada tabel tersebut.

2.6.7 Konsep Relasi

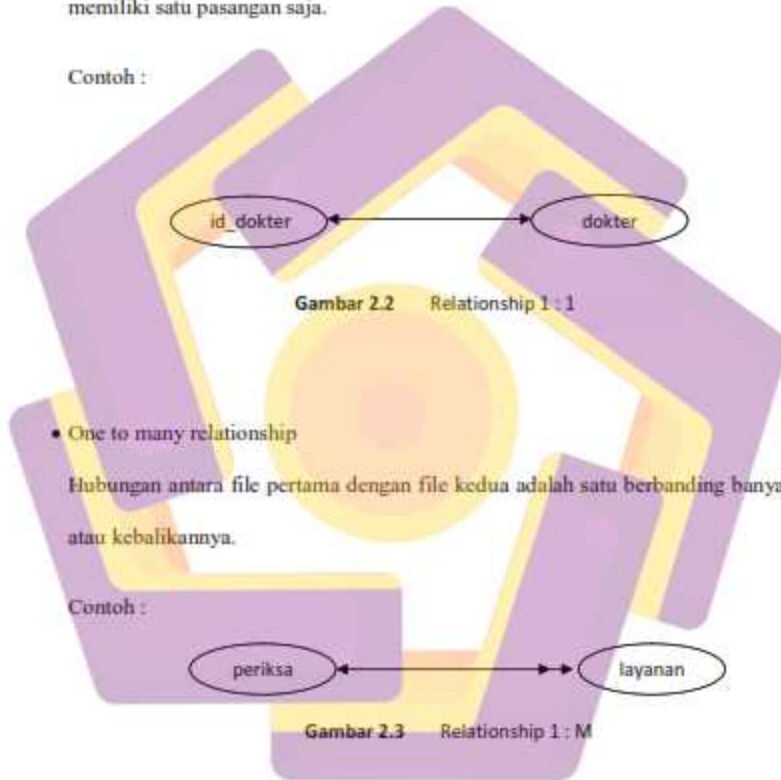
Relasi berarti hubungan atau keterkaitan. Dalam hal perancangan database relasi diartikan sebagai hubungan atau keterkaitan antara sebuah atau beberapa entitas dengan sebuah atau beberapa entitas lainnya.

Dalam database ada tiga konsep relasi yang pemakaiannya dapat dikombinasikan sesuai dengan konteks data. Adapun konsep relasinya adalah sebagai berikut :

- One to one relationship

Hubungan antara file pertama dengan file kedua adalah berbanding satu atau memiliki satu pasangan saja.

Contoh :



- One to many relationship

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau kebalikannya.

Contoh :



- Many to many relationship

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak.

Contoh :



Gambar 2.4 Relationship M : M

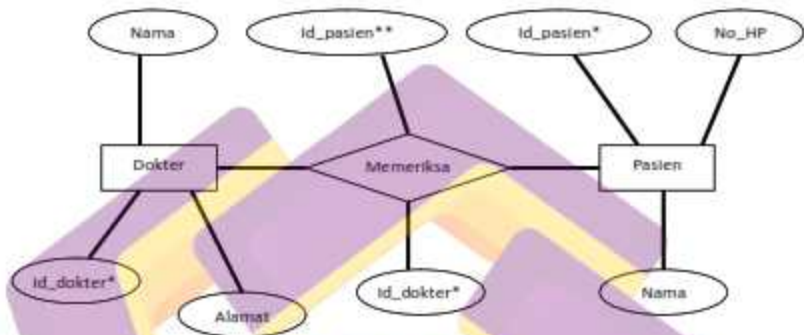
- Entity relationship model

Entity adalah pemodelan data yang berdasarkan pada objek dasar yang benar-benar nyata, relationship adalah hubungan antar objek, sedangkan atribut merupakan sebutan untuk mewakili suatu entity. Atribut dalam ERD dilambangkan dengan bentuk elips. Untuk lebih jelasnya dapat dilihat pada gambar 2.5.



Gambar 2.5 Entity Relationship

Contoh entity relationship dalam sebuah himpunan relasi memeriksa antara Dokter dan Pasien :



Gambar 2.6 Relasi Mengajar

Ada beberapa istilah yang harus dipahami sebelum merancang basis data dengan menggunakan model entity relationship diagram, diantaranya yaitu :

1. Superkey

Satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam tabel secara unik.

Sebagai ilustrasi diberikan tabel pasien berikut :

Tabel 2.3 Tabel Pasien

Pasien

Id_Pasien	Nama	Alamat	JK
1001	Rita	Klaten	P
1002	Rafa		L
1003	Rita	Jogja	P
1004	Vanessa		P

Dari tabel tersebut yang mungkin menjadi superkey adalah atribut Id_Pasien, atribut Id_Pasien dan Nama, atribut Id_Pasien dan Alamat, atribut Id_Pasien dan JK, atribut Id_Pasien, Nama dan Alamat, atribut Id_Pasien, Nama, dan JK, atribut Id_Pasien, Nama, Alamat dan JK.

Atribut Nama tidak bisa menjadi superkey karena memungkinkan lebih dari 1 baris yang memiliki nilai sama. Demikian pula atribut alamat, atribut JK, atribut Nama dan Alamat, atribut Nama dan JK, atribut Alamat dan JK serta atribut Nama, Alamat dan JK.

2. Kandidat Key

Merupakan kumpulan atribut minimal yang membedakan setiap baris data dalam tabel secara unik. Untuk bisa menjadi kandidat key, suatu atribut harus memenuhi persyaratan sebagai berikut :

- a. Untuk satu nilai hanya mengidentifikasi satu baris dalam satu relasi (unik)
- b. tidak memiliki subset yang juga merupakan kunci relasi
- c. tidak dapat bernilai null

Dengan kata lain, sebuah kandidat key adalah superkey yang paling sedikit jumlah atributnya. Dari contoh pada superkey dari tabel Pasien pada Tabel 2.3, yang memenuhi persyaratan sebagai kandidat key adalah atribut Id_Pasien.

3. Kunci Relasi (*Relation Key*)/Kunci Utama (*Primary Key*)

Nilai dari kunci relasi harus mengidentifikasi sebuah baris yang unik di dalam sebuah relasi. Agar bisa menjadi sebuah *primary key*, sebuah atribut haruslah memenuhi persyaratan sebagai kandidat key. Dalam contoh tabel pasien pada Tabel 2.3, Id_Pasien adalah kunci relasinya.

4. Kunci Alternatif

Kunci yang tidak ada di dunia nyata, tetapi diadakan dan dijadikan *primary key*. Kunci alternatif dibuat ketika tidak ada satu pun atribut dalam sebuah relasi yang bisa mewakili relasi tersebut, atau ada yang bisa menjadi kandidat key tetapi tidak cukup efektif untuk digunakan sebagai *primary key*.

Sebagai contoh berikut adalah sebuah relasi Layanan :

Tabel 2.4 Tabel Layanan

Layanan

Nama	Keterangan
Cabut Gigi	
Suntik	Yang nusukin jarum itu lho..
Rontgen	
Tes Narkoba	Buat ngetes pake narkoba ato gak gitu dech,

Dari Tabel 2.4, kandidat key pada tabel tersebut adalah Nama. Tetapi untuk lebih efektif mungkin saja perancang basis data akan memberikan kunci alternatif berupa kode_layanan. Kode tersebut tidak pernah ada dalam dunia nyata, karena kita tidak pernah tahu kalau Suntik itu kodenya I, atau Rontgen itu kodenya R. Tetapi kode ini diadakan dan dipilih sebagai kunci utama, maka kode_layanan disebut sebagai kunci alternatif.

5. Foreign Key (FK)

Istilah FK juga banyak digunakan dalam perancangan. Sebuah FK adalah sekumpulan atribut dalam satu relasi (misal A) sedemikian sehingga kumpulan atribut ini bukan kunci relasi A tetapi merupakan kunci dari relasi lain.

2.7 PHP dan MySQL

2.7.1 Mengetahui PHP

Bahasa pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah web server. Script-script PHP yang dibuat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut. Penggunaan program PHP memungkinkan sebuah website menjadi lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung website akan diolah dan disimpan dalam database web server dan bisa ditampilkan kembali apabila diakses.

Berikut beberapa keunggulan yang dimiliki program PHP :

- PHP bersifat *free* atau gratis
- Beberapa server seperti Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd, dan Xitami mampu menjalankan PHP
- Tingkat akses PHP lebih cepat serta memiliki tingkat keamanan yang tinggi
- Beberapa database yang sudah ada, baik yang bersifat *free/gratis* ataupun komersial sangat mendukung akses PHP, diantaranya MySQL, PostgreSQL, mSQL, Informix, dan MicrosoftSQL server.

- PHP mampu berjalan di Linux sebagai platform sistem operasi utama bagi PHP, tetapi dapat juga berjalan di FreeBSD, Unix, Solaris, Mac, Windows, dan yang lainnya.

Script PHP memiliki struktur dalam penulisannya. Script PHP biasanya diapit oleh tanda sebagai berikut :

- `<?php ... code program ... ?>`
- `<? ... code program ... ?>`
- `<script language="PHP"> ... code program ... </script>`

2.7.1.1 Operator Milik PHP

Operator atau tanda yang biasanya digunakan untuk sebuah proses. Ada beberapa operator dalam PHP yaitu *Operator Aritmatika*, *Operator Assignment*, *Operator Perbandingan*, dan *Operator Logika*. Perhatikan tabel di bawah ini untuk keterangan masing-masing operator tersebut.

Tabel 2.5 Tabel Operator dalam PHP

No	Nama / Simbol Operator	Fungsi
1.	Operator Aritmatika	
	+	Digunakan untuk penjumlahan, contoh : Stambah = 30 + 40;

	-	Untuk pengurangan, contoh : $Skurang = 40 - 30$;
	*	Melakukan perkalian, contoh : $Skali = 30 * 40$;
	/	Melakukan pembagian, contoh : $Sbagi = 30 / 40$;
	%	Mendapatkan sisa hasil pembagian, contoh : $Ssisa_bagi = 10 \% 3$; akan menghasilkan 1.
2. Operator Assigment		
	=	Memberi nilai suatu variabel, contoh : $Snilai = 10$;
	++	Menambahkan nilai sebanyak n data, contoh : $Snilai = 10$; $Snilai ++ 5$; maka isi variabel $Snilai$ sekarang adalah 15.
	--	Mengurangkan nilai sebanyak n data, contoh : $Snilai = 10$; $Snilai -- 5$; maka isi variabel $Snilai$ sekarang adalah 5.
	*=	Mengalikan nilai sebanyak n data, contoh : $Snilai = 10$; $Snilai *= 5$; maka isi variabel sekarang adalah 50.

	/=	<p>Membagikan nilai sebanyak n data, contoh :</p> <p>Snilai = 10;</p> <p>Snilai /= 5; maka isi variabel sekarang adalah 2.</p>
3. Operator Perbandingan		
	==	<p>Akan bernilai 1 (true) bila data yang dibandingkan bernilai sama, contoh :</p> <p>Sdata1 = 10; Sdata2 = 10;</p> <p>Sdata1 == Sdata2, nilai true, nilainya sama-sama 10.</p>
	!=	<p>Bernilai 1 (true) bila data yang dibandingkan bernilai tidak sama atau 0 (false) bila ternyata datanya bernilai sama, contoh ;</p> <p>Sdata1 = 10; Sdata2 = 10;</p> <p>Sdata1 != Sdata2, nilai false, nilainya sama-sama 10.</p>
	>	<p>Bernilai true bila data sebelah kiri Lebih Besar dari data yang sebelah kanan, contoh :</p> <p>Sdata1 = 12; Sdata2 = 10;</p> <p>Sdata1 > Sdata2, nilai true karena Sdata1 nilainya lebih besar dari Sdata2.</p>
	<	<p>Bernilai true bila data sebelah kiri Lebih Kecil dari data yang</p>

		<p>sebelah kanan, contoh :</p> <p><code>Sdata1 = 12; Sdata2 = 10;</code></p> <p><code>Sdata1 > Sdata2</code>, nilai false karena <code>Sdata1</code> nilainya lebih besar dari <code>Sdata2</code>.</p>
	<p>↳</p>	<p>Bernilai true bila data sebelah kiri Lebih Besar Sama Dengan dari data yang sebelah kanan, contoh :</p> <p><code>Sdata1 = 12; Sdata2 = 10;</code></p> <p><code>Sdata1 >= Sdata2</code>, nilai true karena <code>Sdata1</code> nilainya lebih besar dari <code>Sdata2</code> atau bila nilai <code>Sdata1 = 10</code> juga akan bernilai true.</p>
	<p>↳</p>	<p>Bernilai true bila data sebelah kiri Lebih Kecil Sama Dengan dari data yang sebelah kanan, contoh :</p> <p><code>Sdata1 = 12; Sdata2 = 10;</code></p> <p><code>Sdata1 <= Sdata2</code>, nilai false karena <code>Sdata1</code> nilainya lebih besar dari <code>Sdata2</code> atau bila nilai <code>Sdata1 = 10</code> maka akan bernilai true.</p>
4.	Operator Logika	
	<code>&&</code>	Dibaca AND, akan bernilai true bila keduanya sama-sama bernilai true, selain itu nilainya false.
	<code> </code>	Dibaca OR, akan bernilai false bila keduanya sama-sama false,

		selain itu bernilai true.
--	--	---------------------------

2.7.1.2 Variabel dan Tipe Data

Variabel digunakan untuk menampung data yang selalu berubah. Data yang disimpan ke dalam variabel memiliki jenis tertentu yang disebut tipe data. PHP mengenal tiga tipe data, yaitu :

1. Integer : tipe data berupa bilangan bulat, misalnya : 1, 13, 17, 27, 4, 1989
2. Double : tipe data berupa bilangan desimal, misalnya : 2.5, 3.14, 22.7 (pemisah berupa titik bukan koma)
3. String : tipe data berupa teks/karakter/huruf. Cara penulisan variabel ini diapit dengan tanda kutip, misalnya : "Amikom", "Komputer".

Aturan penulisan variabel dalam PHP adalah :

1. Selalu diawali dengan tanda \$ (dollar) diikuti dengan nama variabelnya, misalnya : \$nama, \$salamat, \$pekerjaan
2. Karakter pertama setelah tanda \$ harus berupa huruf atau garis bawah baru setelahnya boleh berupa angka, misalnya : \$umur, \$id1, \$_1
3. Untuk nama variabel yang lebih dari satu kata harus disambung dengan tanda garis bawah bukan spasi, misalnya : \$nama_lengkap, \$jenis_kelamin.

PHP tidak menghususkan pemberian nama variabel sesuai dengan tipe datanya, misalnya : \$nilai=90, bisa juga \$nilai="Sembilan Puluh". Jadi sebuah variabel yang seharusnya angka dapat diisi dengan tipe data string atau huruf. Selain itu variabel dapat diberi nilai kapan saja saat dibutuhkan dan bahkan dapat mengubahnya dengan tipe data berbeda.

PHP mengenal hukum *case sensitive*, artinya variabel akan dibedakan sesuai dengan besar kecilnya huruf, contoh \$nama akan berbeda dengan \$Nama atau \$NAMA.

2.7.2 Mengenal MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (<http://www.wikipedia.org>).

Terdapat beberapa API (*Application Programming Interface*) tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai

bahasa pemrograman untuk dapat mengakses basis data MySQL, antara lain :
bahasa pemrograman C, C++, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python,
Ruby, REALbasic dan Tcl.

MySQL memiliki beberapa keistimewaan, antara lain :

1. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, Mac OS, FreeBSD, dan masih banyak lagi
2. MySQL bersifat *Open Source* di bawah lisensi GPL, sehingga dapat digunakan secara gratis
3. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double, char text, date, timestamp, dan lain-lain
4. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem password yang mendetail serta sandi terenkripsi
5. MySQL mampu menangani basis data dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya
6. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa, namun sayang sampai tulisan ini ada bahasa Indonesia belum termasuk di dalamnya
7. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Kehandalan suatu sistem basis data dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna atau program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional ataupun operasi basis data non-transaksional. Pada modul non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya. Namun demikian dalam modul non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modul non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti blogging berbasis web, cms, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modul basis data transaksional, hanya saja sebagai konsekuensinya untuk kerja MySQL pada modul transaksional tidak secepat unjuk kerja pada modul non-transaksional.

2.8 NowSMS

Dalam pembuatan sebuah sistem informasi yang berbasis SMS, aplikasi NowSMS bisa dikatakan sebagai "Sang Aktor Utama". Karena komponen inilah yang menjembatani pentransferan data-data SMS dari handphone/mobile modem ke komputer atau sebaliknya.

NowSMS dan MMS Gateway adalah sebuah produk *software sever* yang menyediakan solusi pengiriman konten SMS dan MMS. NowSMS adalah SMS

Sever, MMS Server, WAP Push Proxy Gateway dan Multimedia Messaging Center (MMSC) yang mudah diinstal. NowSMS merupakan solusi jelas yang terjangkau untuk pengembangan dan pengujian dengan skalabilitas untuk mendukung produksi penuh sistem operator handphone.

NowSMS dan MMS Gateway adalah alat yang sangat kuat dan fleksibel dengan lebih dari 1500 instalasi yang dibeli oleh pelanggan dari 6 benua dan puluhan pulau di sekitarnya. Tapi kebanyakan pengguna NowSMS tidak menggunakan semua fitur NowSMS, mereka hanya menggunakan satu atau beberapa dari banyak fitur yang diberikan NowSMS.

Perlu diperhatikan bahwa NowSMS bukanlah penyedia layanan SMS atau MMS masal. NowSMS juga bukanlah pengganti operator penyedia layanan SMS atau MMS. Perlu koneksi ke penyedia layanan SMS atau modem GSM sebelum mengirim pesan apapun dengan NowSMS (<http://www.nowSMS.com>).



Gambar 2.7 Tampilan menu aplikasi NowSMS