

BAB II

LANDASAN TEORI

2.1 Studi Kepustakaan

Penulis mengambil beberapa referensi kepustakaan yang bersumber pada penelitian-penelitian sebelumnya. Hal ini berguna sebagai perbandingan bahan referensi dalam menyelesaikan penelitian ini.

Arief Sulaiman Fathoni (2014) menjelaskan dalam penelitiannya bahwa proses *monitoring* pelayanan pelanggan PT PLN Distribusi Jatim Area Ngawi masih dilakukan secara manual. Selama ini pokok permasalahan dalam kinerja pelayanan penyambungan salah satunya adalah sistem *monitoring* yang masih ditulis dalam buku agenda sehingga masih banyaknya keluhan pelanggan terhadap lama penyambungan yang menuntut perlunya peningkatan pengawasan terhadap kinerja petugas penyambungan, yakni suatu bentuk *monitoring* yang mudah dan praktis dalam mengakomodir *volume* penyambungan yang relatif tinggi di area Ngawi. Penulis kemudian mengembangkan sistem yang dapat meningkatkan kinerja pelayanan penyambungan melalui kepastian waktu pelayanan pasang baru perubahan daya dan meningkatkan kepuasan pelanggan. Sistem ini dibangun dengan bahasa pemrograman PHP dan MySQL sebagai *database*.

Sistem ini memiliki persamaan dengan sistem yang akan penulis bangun. Keduanya merupakan sistem *monitoring* yang dapat mempermudah petugas dalam proses pengawasan. Bahasa pemrograman yang dipakai juga sama yakni PHP serta MySQL sebagai *database*. Namun demikian, keduanya memiliki perbedaan yang signifikan. Sistem sebelumnya diterapkan PT. PLN Distribusi

Jatim Area Ngawi, sementara sistem yang akan penulis bangun diterapkan pada PT. Sinar Mandiri Alam Sejahtera untuk *monitoring* gas LPG. Sehingga dipastikan banyak perbedaan dalam penggunaannya. Selain itu, sistem yang akan penulis bangun juga sangat lengkap. Pertama adalah fitur pendaftaran sebagai agen gas LPG, kedua adanya fitur LBS yang bisa digunakan petugas untuk survei lokasi, *SMS Gateway* untuk proses notifikasi, *QR Code* sebagai bukti pengambilan gas, dan fitur *monitoring* oleh petugas.

Gumilang, Febriyansyah (2007) dalam penelitiannya menjelaskan bahwa Sistem Informasi Pelayanan IMB di Unit Pelayanan Terpadu (UPT) Pemerintah Daerah Semarang masih menggunakan cara manual. Cara ini tidak sesuai dengan perkembangan teknologi yang sudah maju pesat. Sistem ini dibangun berbasis *Computer Telephony Integration* dan bahasa pemrograman dengan menggunakan Delphi. Penelitian ini dapat memberikan informasi yang dibutuhkan oleh pemohon IMB, meliputi syarat, prosedur, status pengurusan (selesai/belum) dan biaya yang harus dikeluarkan.

Sistem ini memiliki persamaan dengan sistem yang akan penulis bangun. Konsepnya adalah sistem pendaftaran yang dapat mempermudah masyarakat ketika akan mendaftar sesuatu. Jika pada penelitian terdahulu masyarakat mendaftar untuk membuat IMB, maka penelitian yang akan dibangun untuk mempermudah masyarakat ketika mendaftar sebagai agen gas LPG. Namun penelitian terdahulu hanya terbatas pada sistem pendaftaran saja. Sementara pada penelitian yang akan penulis bangun sistemnya sangat lengkap. Tidak hanya terbatas pada sistem pendaftaran saja, namun dilengkapi dengan banyak fitur

lainnya seperti fitur pendaftaran sebagai agen gas LPG, adanya fitur LBS yang bisa digunakan petugas untuk survei lokasi, *SMS Gateway* untuk proses notifikasi, QR Code sebagai bukti pengambilan gas, dan fitur *monitoring* oleh petugas. Selain itu, sistem ini dirancang berbasis *mobile* dengan *framework jQuery* serta bahasa pemrograman PHP dan *database MySQL*. Sedangkan penelitian terdahulu masih menggunakan *Computer Telephony Integration* dan bahasa pemrograman menggunakan Delphi.

Eva Kurniawaty (2012), dalam penelitiannya menjelaskan bahwa proses perizinan *online* pada Dinas Perindustrian dan Perdagangan Kota Surabaya masih dilakukan secara manual. Masyarakat harus datang langsung dengan membawa persyaratan dalam bentuk *hardcopy*. Mereka harus mengatase terlebih dahulu untuk menunggu giliran melakukan pendaftaran. Sistem ini tidak sesuai dengan perkembangan teknologi yang sangat pesat seperti saat ini. Sistem pendaftaran *online* berbasis *web* dan dibangun dengan menggunakan teknologi PHP, sedangkan untuk *database* menggunakan MySQL. Sistem ini mempermudah sistem yang sebelumnya manual menjadi sistem yang terkomputerisasi. Dengan sistem ini, pemohon dapat melakukan pendaftaran tentang perizinan legalitas usaha baik bidang industri maupun non industri dimana saja dan kapan saja. *User* melakukan registrasi terlebih dahulu untuk *login* ke sistem, kemudian mengisi data untuk mendapat nomor masuk untuk *user*, baru kemudian *user* melengkapi persyaratan persyaratan yang telah ditentukan. Setelah itu petugas kembali mengecek data permohonan yang telah dilengkapi, dan meneruskan ke pihak

manajemen untuk memberikan persetujuan. Selesai disetujui, petugas memberikan nomor keluar untuk SK.

Secara konsep, sistem ini memiliki persamaan dengan sistem yang akan penulis bangun. Dalam satu fiturnya terdapat sistem yang dapat mempermudah *user* dalam melakukan suatu pendaftaran. Bahasa pemrograman yang digunakan juga sama yakni PHP dan *database* menggunakan MySQL. Namun keduanya memiliki perbedaan yang sangat jauh. Dimana pada sistem sebelumnya dibangun masih berbasis *web*, sementara sistem yang akan penulis bangun sudah berbasis *mobile*. Sistem ini juga lebih lengkap dibanding sebelumnya. Tidak hanya terbatas pada sistem pendaftaran saja, namun dilengkapi dengan banyak fitur lainnya seperti fitur pendaftaran sebagai agen gas LPG, adanya fitur LBS yang bisa digunakan petugas untuk survei lokasi, *SMS Gateway* untuk proses notifikasi, *QR Code* sebagai bukti pengambilan gas, dan fitur monitoring oleh petugas.

2.2 Dasar Teori

2.2.1 Sistem

Sistem merupakan kumpulan dari unsur-unsur atau elemen-elemen yang membentuk suatu kesatuan dan saling bekerjasama untuk mencapai suatu tujuan. Sistem juga dapat diartikan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu: komponen sistem, batas sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolah sistem, dan sasaran sistem. (Jogiyanto, 2005)

2.2.2 Sistem Monitoring

Sistem *monitoring* merupakan sistem yang didesain untuk bisa memberikan *feedback* ketika program sedang menjalankan fungsinya. *Feedback* dimaksudkan untuk memberikan informasi keadaan sistem pada saat itu. Sistem *monitoring* juga dapat diartikan sebagai kumpulan prosedur dan program untuk mengkomputasi sistem informasi yang didesain untuk mencatat dan mentransmisikan data berdasarkan informasi yang diperoleh. Selain itu sistem *monitoring* adalah kumpulan fitur informatif yang memberikan informasi mengenai apa saja yang terjadi dengan sistem yang di-*monitor*. (Siswanto, 2014)

2.2.3 Aplikasi Mobile

Aplikasi *Mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya.

Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, desain dan *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop, yaitu antara lain :

1. Ukuran yang Kecil

Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memori yang Terbatas

Perangkat *mobile* juga memiliki *memory* yang kecil, yaitu *primary* (RAM) dan *secondary* (*disk*). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari *memory*, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

3. Daya Proses yang Terbatas

Sistem *mobile* tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti *hard disk* dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

4. Mengonsumsi Daya yang Rendah

Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

5. Kuat dan Dapat Diandalkan

Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Konektivitas yang Terbatas

Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi *wireless*.

7. Masa Hidup yang Pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka *booting* dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol *power* mereka. (Buyens, 2010)

2.2.4 SMS Gateway

Istilah gateway dapat diartikan sebagai pintu gerbang Namun pada dunia komputer, gateway dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem yang lain. Dengan demikian, *SMS Gateway* dapat diartikan sebagai penghubung untuk lalu lintas data-data SMS. (Yunianto, 2006)

Pada awalnya *SMS Gateway*, dibutuhkan untuk menjembatani antara SMSC. Hal ini dikarenakan SMSC (*Short Message Service Center*) yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri, dan protokol tersebut bersifat pribadi. *SMS Gateway* ini kemudian ditempatkan diantara kedua SMSC yang berbeda pada protokol tersebut, yang akan menerjemahkan data dari SMSC satu ke protokol SMSC lainnya yang dituju.

Namun seiring perkembangan teknologi komputer dan teknologi komunikasi, *SMS Gateway* tidak lagi dimaksudkan sebagaimana pada penjelasan diatas.

Sekarang masyarakat mengartikan *SMS Gateway* sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel). *SMS Gateway* kemudian lebih mengarah kepada sebuah program yang mengkomunikasikan sistem operasi komputer dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS. Cara kerja *SMS Gateway* pada dasarnya hampir sama dengan mengirimkan SMS melalui *handphone* pada umumnya. Hanya saja, bedanya adalah perangkat pengirimnya bukan lagi *handphone* tetapi modem GSM. Dan modem inilah yang dikendalikan oleh PC menggunakan aplikasi *SMS Gateway* yang akan dibuat. 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.
- (Fikri, 2012).

2.2.5 Location Based Services (LBS)

Location Base Services (LBS) suatu layanan yang memberikan informasi berdasarkan posisi pengguna (*device*) sebagai acuan utama. Saat ini LBS digunakan dalam berbagai konteks seperti kesehatan, pencarian objek, hiburan, pekerjaan dan lain-lain. dua unsur utama dalam LBS yaitu :

1. *Location Manager (API Maps)*

Menyediakan perangkat bagi sumber atau source untuk LBS, *Application Programming Interface* (API) menyediakan fasilitas untuk menampilkan atau memanipulasi peta.

2. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang di gunakan oleh perangkat. API Location berhubungan dengan data GPS dan data lokasi *real-time*. Data lokasi pengguna biasanya di dapatkan melalui jaringan telepon seluler ataupun menggunakan GPS. LBS memiliki komponen-komponen yang menunjang dalam prosesnya yaitu :

- a. Perangkat mobile. Pengguna membutuhkan perangkat *mobile* untuk menggunakan layanan LBS ini. Sepertihalnya *smartphone*, *tablet* dan lain-lain.
- b. Jaringan komunikasi. Jaringan komunikasi digunakan untuk menghubungkan perangkat *mobile* dengan perangkat lainnya
- c. Komponen pengambil posisi latitude dan longitude (satelit). Satelit merupakan alat yang menentukan posisi pengguna. Seperti jarak, lokasi dan lain-lain
- d. Data dan *provider content*. Data yang di dapatkan akan di proses di server dan dikirim kembali ke pengguna berupa data yang telah akurat
- e. WMS (*web map server*). Merupakan *server* dimana tempat pengumpulan dan pemrosesan data

Berikut ini adalah cara kerja dari LBS :

1. Pertama *smartphone* membuka aplikasi LBS yang sudah terinstall / jika menggunakan aplikasi yang berbasis *browser*, maka buka *browser* dan ketik alamat tujuan situsnya.
2. Aplikasi LBS akan melakukan sambungan dengan jaringan *provider* (seperti telkomsel, xl, axis, dll) yang dipakai oleh si *User* (pengguna).
3. Jaringan mengirimkan *request* ke satelit untuk menentukan longitude (garis bujur) dan latitude (garis lintang) dari si pengguna aplikasi tersebut.
4. Provider menghubungkan aplikasi (di *smartphone*) dengan *server* LBS dan meminta data yang diinginkan *User*.
5. *User* mendapatkan data dan ditampilkan di *smartphone*. (B.R.Rompas, 2012)

2.2.6 Google Maps API

Google Maps adalah peta *online* atau membuka peta secara *online*, dapat dilakukan secara mudah melalui layanan gratis dari *Google*. Bahkan layanan ini menyediakan API (*Application Programming Interface*) yang memungkinkan *developer* lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Tampilan *Google Maps* pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.

Google Maps adalah layanan gratis yang diberikan oleh *Google* dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur

Google Maps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi Internet yang sangat stabil.

Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik *Google* sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. Dalam pembuatan program *Google Map API* menggunakan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML.
2. Membuat element div dengan nama `map_canvas` untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event `onload`.

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. ROADMAP, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi.

2. SATELLITE, untuk menampilkan foto satelit.
3. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
4. HYBRID, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota). (Nazarudin, 2013).

2.3 Alat Bantu Dalam Analisa dan Perancangan Sistem

2.3.1 *Data Flow Diagram* (DFD)

Diagram yang menggunakan notasi-notasi ini untuk menggambarkan arus dari *data* sistem sekarang dikenal dengan nama diagram arus *data* (*data flow diagram* atau DFD). DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana *data* tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana *data* tersebut akan disimpan (*file* kartu, *microfiche*, *hard disk*, *tape*, *diskette* dan lain sebagainya).

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang (*structured Analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus *data* di dalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD juga merupakan dokumentasi dari sistem yang baik. (Jogiyanto, 2005).

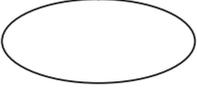
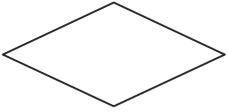
Tabel 2.1 Simbol *Data Flow Diagram*

Simbol	Nama	Fungsi
	Simbol entitas eksternal	Digunakan untuk menunjukkan tempat asal <i>data</i> atau sumber atau tempat tujuan <i>data</i> atau tujuan.
	Simbol proses	Digunakan untuk menunjukkan tugas atau proses yang dilakukan baik secara manual atau otomatis
	Simbol penyimpanan <i>data</i>	Digunakan untuk menunjukkan gudang informasi atau <i>data</i> .
	Simbol arus <i>data</i>	Digunakan untuk menunjukkan arus dari proses.

2.3.2 *Entity Relationship Diagram (ERD)*

ERD merupakan diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada sebuah *database* sistem.

Tabel 2.2 Simbol *Entity Relationship Diagram*

Simbol	Nama	Keterangan
	Simbol <i>entity</i>	Suatu <i>entity</i> merupakan suatu objek atau konsep mengenai tempat yang anda inginkan untuk menyimpan informasi
	Simbol atribut	Atribut adalah sifat-sifat atau karakteristik suatu entitas
	Simbol Relasi	Relasi mengilustrasikan bagaimana dua entitas terbagi informasi didalam struktur basis data

Konektivitas dari suatu hubungan menguraikan pemetaan dari kejadian entitas yang dihubungkan. Jenis dasar konektivitas untuk hubungan adalah satu-ke-satu, satu-ke-banyak, dan banyak-ke-banyak. (Yakub, 2012).

Tabel 2.3 Simbol Kardinalitas Dalam *Entity Relationship Diagram*

Simbol yang Digunakan	Jenis Kerelasiaan
	1-ke-1
	1-ke-n
	n-ke-1
	n-ke-n

2.3.3 Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analis sistem atau pemrogram dapat mengikuti pedoman-pedomain sebagai berikut

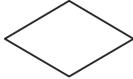
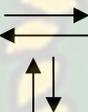
1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir

4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus tunjukkan dengan jelas menggunakan simbol penghubung
7. Gunakan simbol-simbol bagan alir yang standar

Selain itu *Flowchart* digunakan untuk mempermudah penyusunan program. Dengan menggunakan *flowchart*, logika pemrograman lebih mudah dipahami dan dianalisis, sehingga anda dapat menentukan kode-kode pemrograman yang sesuai dengan pekerjaannya. *Flowchart* program dapat disamakan dengan *blue print* bangunan. Seperti diketahui arsitek akan membuat *blue print* bangunan sebelum memulai konstruksinya. Demikian pula seorang *programmer* disarankan untuk membuat *flowchart*. Sebelum menulis kode programnya. Berikut beberapa simbol standar *flowchart* yang sering digunakan dalam pemrograman komputer. (Jogiyanto, 2005).

Tabel 2.4 Simbol *Flowchart*

Simbol	Fungsi
	Simbol <i>Start</i> atau <i>End</i> yang mendefenisikan awal atau akhir dari sebuah <i>flowchart</i>
	Simbol pemrosesan yang terjadi pada sebuah alur kerja
	Simbol <i>Input/output</i> mendefenisikan masukan dan keluaran proses

	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
	Simbol konektor untuk menyambung proses lembar kerja yang berbeda
	Simbol untuk menghubungkan antar proses atau antar simbol
	Simbol yang menyatakan bagian dari program (sub program)

2.4 Alat Bantu Dalam Pembuatan Sistem

2.4.1 HTML

Hypertext Markup Language adalah salah satu format yang digunakan untuk menulis halaman web, HTML ini berjalan di *web browser* dan memiliki fungsi untuk melakukan pemrograman aplikasi berbasis *web*. HTML merupakan bahasa (aturan) standar yang digunakan untuk menampilkan teks, gambar, video dan audio ke dalam halaman web. Selain itu HTML juga merupakan *file* teks yang tersusun atas elemen-elemen yang disebut dengan *tag*. (Raharjo, 2011)

2.4.2 PHP

PHP (*hypertext preprocessor*) merupakan bahasa script yang ditanam di sisi *server*. PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman *web* yang dinamis, serta menerima dan menciptakan cookies. PHP dapat berjalan pada semua jenis sistem operasi antara lain pada Linux dan varian Unix (HP-UX,

Solaris dan OpenBSD), pada Ms Windows, Mac dan masih banyak lagi, selain itu PHP juga dapat berjalan pada beberapa jenis *web-server* antara lain Apache, Microsoft IIS, personal *webserver*, Netscape dan lain sebagainya. Adapun keunggulan dari PHP adalah sebagai berikut:

1. Kesederhanaan

Buat Anda yang baru belajar pemrograman alasan ini merupakan alasan pertama untuk mulai belajar PHP. Umumnya bahkan yang sedikit tahu atau bahkan sama sekali yang tidak tahu tentang pemrograman bisa dengan cepat dan mulai membuat aplikasi lengkap. Hal ini memungkinkan karena PHP khusus didesain untuk membuat aplikasi web dan karena itu, PHP memiliki banyak sekali fungsi build in untuk memenuhi kebutuhan standar aplikasi *web*.

2. PHP Bersifat *Open Source*

Karena *source code* PHP tersedia secara gratis, maka hal itu memungkinkan komunitas *developer* untuk melakukan perbaikan, pengembangan, dan menemukan *bug* dalam bahasa PHP.

3. Stabilitas dan Kompabilitas

Saat ini PHP berjalan dengan stabil diberbagai macam sistem operasi seperti berbagai versi UNIX (Linux), Windows, dan Mac. PHP juga terintegrasi secara baik dengan berbagai macam *web server* termasuk dua yang terpopuler ISS dan Apache.

4. *Support Terhadap Semua Database*

PHP juga dilengkapi dengan pendukung lain seperti support langsung ke berbagai macam *database* yang populer, arsitektur yang dapat dikembangkan dan prosesor yang hanya menggunakan resource minimal komputer, tetapi juga dapat menampilkan halaman *web* dengan cepat. (Prasetio, 2014)

2.4.3 *JQuery Mobile*

JQuery Mobile adalah sebuah *framework* berbasis *JavaScript*. *JQuery* sama dengan *JavaScript Library* yaitu kumpulan kode atau fungsi *JavaScript* siap pakai sehingga mempermudah dan mempercepat kita dalam membuat kode *JavaScript*. Hal yang menarik dari *JQuery* penekanan interaksi antara *JavaScript* dan HTML. *JQuery* pertama kali dirilis pada tahun 2006 oleh John Resig. *JQuery* pertama kali dirilis pada tahun 2006 oleh John Resig. *JQuery* memiliki slogan "Write less, do more" yang artinya kesederhanaan dalam penulisan kode, tapi dengan hasil yang lebih banyak. Selanjutnya *JQuery* akan berkembang lebih lanjut untuk perangkat mobile yang dinamakan *JQuery Mobile*.

Dengan kemudahan yang ada pada *JQuery Mobile* akan sangat membantu dalam membuat sendiri aplikasi untuk web mobile phone. Aplikasi ini akan membuat mobile phone anda menjadi lebih asyik digunakan. Adapun kemampuan atau fitur yang dimiliki *Library JQuery* antara lain sebagai berikut

1. Memudahkan pengembangan *user-interface* untuk *mobile web apps*.
2. Bersifat *markup-driven*, sehingga dengan *markup* standar, bisa menggunakan *jquery mobile* tanpa membuat koding HTML sendiri.

3. Bisa mengakomodasi beberapa *event* untuk perangkat *mobile*, seperti bisa mendeteksi *event* perangkat *touchscreen* seperti *tap*, *tap-and-hold*, *swipe*, dan adanya perubahan orientasi (misalnya merotasi piranti).
4. Memastikan bahwa *user interface* bekerja pada *web browser*.
5. Memiliki beberapa *theme* yang memudahkan perubahan tampilan aplikasi. (Tim E-Media Solusindo, 2013)

2.4.4 JavaScript

JavaScript merupakan bahasa pemrograman berbasis *client*, artinya bahasa ini berjalan pada sisi browser pengguna, bukan pada *server*. Salah satu keuntungan dari bahasa ini adalah ringan karena berjalan pada masing-masing *browser* dan pekerjaan tidak dibebankan pada server. Jika diakses oleh 1000 orang maka setiap beban akan ditanggung oleh masing-masing *browser*, bukan *server* yang menanggung eksekusi 1000 pengakses tersebut.

Selain itu *JavaScript* adalah bahasa skrip yang digunakan untuk mengontrol tindakan-tindakan yang diperlukan di halaman *web*. Sebagai contoh, ketika pengguna mengarahkan penunjuk *mouse* ke suatu gambar maka gambar bisa diatur melalui kode *JavaScript*. Contoh lain, kode *JavaScript* dapat dipakai untuk memeriksa isi suatu *field* teks sudah diisi oleh pengguna atau belum. (Kadir, 2013).

2.4.5 CSS

CSS adalah suatu bahasa *stylesheets* yang mengatur tampilan suatu dokumen. Pada umumnya CSS digunakan untuk mengatur tampilan dokumen.

CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda. Dengan CSS, tampilan website akan lebih cantik dan konsisten. Ada dua cara untuk menuliskan kode CSS. Pertama secara internal, yaitu menuliskan langsung diantara tag HTML/XHTML. Kedua secara eksternal, yaitu kode CSS disimpan dalam file yang terpisah kemudian dipanggil saat halaman web dibuka, CSS sendiri merupakan sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium (W3C) dan diperkenalkan pada tahun 1996. (Sulistiyawan, Rubianto, Saleh, 2008)

2.4.6 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan standar SQL (*Structured Query Language*). Dengan menggunakan MySQL *server* maka data dapat diakses oleh banyak pemakai secara bersamaan sekaligus dapat membatasi akses para pemakai berdasarkan *privillage* (hak *user*) yang diberikan. MySQL menggunakan bahasa SQL (*Structure Query Language*) yaitu bahasa standar pemrograman *database*. Adapun keunggulan dari MySQL adalah sebagai berikut:

1. Kecepatan

Dengan adanya fitur tambahan *Query Chacing* pada MySQL yang pada umumnya akan menaikkan kinerja diatas rata-rata, Fitur ini menjadi kelebihan dibandingkan pendahulunya.

2. Open Source

Dengan konsep ini siapapun dapat berpartisipasi mengembangkan MySQL dan hasil pengembangan itu diserahkan kepada umum atau kepada komunitas *Open Source*.

3. Kapabilitas

MySQL telah digunakan untuk mengelola *database* dengan jumlah 50 juta record. Bahkan sanggup untuk mengelola 60.000 tabel.

4. Konektifitas dan Keamanan

MySQL menerapkan sistem keamanan dan ijin akses tingkat lanjut, termasuk dukungan keamanan dengan pengacakan lapisan data. Adanya tingkatan *user* dan jenis akses yang beragam dapat memberikan batasan akses berdasarkan lokasi pengaksesan dan dapat jaminan keamanan pada MySQL.

5. Fleksibilitas dan Portabilitas

MySQL mendukung perintah-perintah ANSI SQL 99 dan beberapa perintah database alternative lainnya sehingga mempermudah beralih dari dan ke MySQL. (Kadir, 2009)

1.