

BAB II

LANDASAN TEORI

2.1 Studi Kepustakaan

Penulis menggunakan acuan yang bersumber pada penelitian-penelitian terdahulu, baik dalam bentuk skripsi, teori, maupun jurnal yang dapat dijadikan sebagai pembandingan serta bahan referensi dalam penyelesaian tugas akhir ini.

Lusiana Sripuspita (2012) dalam penelitiannya menjelaskan bahwa cara pendaftaran tanah milik adat di Badan Pertanahan Nasional Kabupaten Garut masih secara manual. Pemohon langsung datang dan menyerahkan syarat-syarat ke Badan Pertanahan Nasional, sehingga terjadi hambatan dalam proses awal pendaftaran sampai pembuatan sertifikat tanah. Selain itu, pembuatan sertifikat tanah milik adat memerlukan waktu yang cukup lama karena banyaknya pemohon dan sistem pencatatan data pemohon menggunakan *Microsoft Excel* yang mengharuskan sehingga banyaknya waktu yang terbuang. Sistem yang dibangun ini mempermudah masyarakat dalam proses pendaftaran tanah milik adat dan pembuatan sertifikat tanah masyarakat. Sistem dibangun dengan berbasis *web* dan menggunakan bahasa pemrograman PHP serta *database MySQL*.

Perbedaan dengan penelitian yang akan dilakukan ialah perancangan sistem yang akan dilakukan lebih lengkap dan kompleks. Terdiri atas fitur *monitoring* oleh atasan dan masyarakat, *SMS Gateway* untuk proses validasi persyaratan dari admin, dan fitur *Location Based Services (LBS)* bagi petugas yang turun ke lapangan.

Faris, Ardianto (2012) dalam penelitiannya menjelaskan bahwa Permasalahan yang dihadapi oleh BPN Kabupaten Pematang adalah dalam menangani arsip dalam

hal penanganan dokumen arsip terjadi ketidak lancar dalam proses administrasi arsip. Untuk dapat mengatasi masalah tersebut, maka diperlukannya sebuah sistem informasi. Dalam sistem informasi tersebut menggunakan metode pengembangan sistem SDLC. Dengan Metode Pengembangan sistem SDLC dapat menyelesaikan masalah pengarsipan data, pembuatan surat tanah dan arsip surat ukur karena memberikan batasan atau lingkup sistem sehingga bisa menentukan perkiraan sumber daya, bisa mengenali berbagai area permasalahan, dapat melakukan penyusunan urutan tugas pekerjaan pengembangan, dan memberikan dasar pengendalian.

Perbedaan dengan sistem yang akan penulis bangun ialah, sistem penulis dirancang dengan *framework jQuery Mobile* sehingga dapat diakses di perangkat *mobile* dan dilengkapi dengan fitur *SMS Gateway* untuk pemberitahuan jika persyaratan sudah divalidasi dan berkas sudah bisa diambil. Selain itu, sistem juga dilengkapi dengan fitur *monitoring* oleh pimpinan dan masyarakat. Pimpinan dapat memantau keseluruhan aktivitas dari BPN, sedangkan masyarakat bisa melihat status surat yang diajukan serta melakukan pengecekan status sertifikat jika ingin membeli tanah. Sistem yang akan dibangun ini juga dilengkapi dengan fitur *Location Based Services (LBS)* yang berfungsi membantu petugas yang bekerja di lapangan agar mudah menemukan lokasi masyarakat.

Ari Ariansah (2010) dalam penelitiannya mengatakan bahwa pelayanan administrasi pertanahan yang diberikan oleh BPN Kota Bandung masih berbelit-belit. Sistem yang berjalan juga memiliki penyimpanan data yang tidak terorganisir dengan baik menyebabkan terjadinya data duplikat atau berulang pada data

pemohon maupun data dokumen lainnya. Permasalahan diatas menyebabkan sulitnya untuk mengetahui data pemilik tanah yang sah sehingga pelayanan yang diberikan oleh pihak BPN kota Bandung menjadi terganggu. Sistem dibangun dengan berbasis web dan menggunakan bahasa pemrograman PHP serta *database* MySQL.

Perbedaan dengan penelitian yang akan dilakukan sistem yang akan dibangun dilengkapi dengan sistem *monitoring* oleh atasan terhadap keseluruhan proses kerja. Selain itu, masyarakat bisa mendaftar dan di validasi oleh petugas dengan informasi disampaikan melalui *SMS Gateway*. Sistem juga dilengkapi dengan fitur *Location Based Services* (LBS) yang berfungsi membantu petugas yang bekerja di lapangan agar mudah menemukan lokasi masyarakat

2.2 Dasar Teori

2.2.1 Pengurusan Hak Tanah

Untuk melakukan pendaftaran hak tanah masyarakat bisa mengambil formulir pendaftaran pada Badan Pertanahan Nasional kemudian mengisi formulir dan melengkapi persyaratan-persyaratan, seperti :

1. Pengisian identitas diri sesuai formulir yang disediakan oleh BPN,
2. fotokopi ktp,
3. fotokopi kartu keluarga,
4. fotokopi pajak bumi dan bangunan,
5. fotokopi surat keterangan ganti kerugian,
6. fotokopi surat pernyataan tidak bersengketa,
7. fotokopi gambar situasi tanah,
8. fotokopi berita acara pemeriksaan atau pengukuran tanah,

9. surat pernyataan dari lurah bahwasannya tanah itu resmi milik pemohon,
10. surat pernyataan riwayat tanah,

kemudian membayar biaya pendaftaran atau PNPB (Penerimaan Negara Bukan Pajak), menyerahkan syarat pendaftaran seperti formulir pendaftaran yang sudah dilengkapi, bukti pembayaran yang sudah di setorkan ke bank penerimaan. Berikutnya pemohon bisa menunggu jadwal pengukuran bidang tanah. Selanjutnya pemohon menunggu instruksi dari petugas BPN jika permohonan diterima, kemudian membayar BPHTB (Bea Perolehan Hak Atas Tanah dan Bangunan) atau Pajak Daerah. Melakukan pendaftaran SK untuk penerbitan sertifikat.

2.2.2 Sistem

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan. Sistem juga dapat diartikan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan 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 (Yakub, 2012).

2.2.3 *Global Positioning System (GPS)*

Global Positioning System (GPS) merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya berada (secara global) di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun posisi saat ini, maka GPS bisa membantu

menunjukkan arah, selama masih terlihat langit. Layanan GPS ini tersedia gratis, bahkan tidak perlu mengeluarkan biaya apapun kecuali membeli GPS *receiver*. (Andi, 2009).

2.2.4 Location Base 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*. Pada dasarnya, LBS terbagi menjadi 2 yaitu :

1. *Pull Service*

Pengguna secara aktif mengirimkan informasi yang dibutuhkan. Sama seperti mengakses sebuah halaman web di browser, kita memasukan halaman web yang di tuju dan kita memperoleh informasi dari halaman

web yang tampil di browser. Pull service terbagi menjadi dua bagian yaitu berdasarkan fungsional seperti memesan taksi dengan menekan tombol pada device atau layanan service mencari lokasi restoran yang terdekat dengan kita.

2. *Push Service*

Memberikan informasi kepada pengguna yang mana tidak secara langsung diminta oleh pengguna. Aplikasi Push Service cocok digunakan untuk memata-matai seseorang melalui SmartPhone yang digunakannya, misalnya : orang tua yang ingin tau anaknya sedang berada dimana, maka dengan adanya aplikasi Push Service ini orang tua bisa mendapatkan Data lokasi anaknya yang dikirim ke Smartphone orang tua yang berasal dari SmartPhone anaknya secara Push Service. sehingga si anak tidak mengetahui kalau orang tua nya sedang memata-matai pergerakan dari si Anak dan menghindari hal-hal yang tidak diinginkan seperti : penculikan, dan lain-lain. (Nazarudin, 2013).

2.2.5 *SMS Gateway*

Gateway menurut kamus Inggris-Indonesia diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* berarti jembatan penghubung antara satu sistem dengan sistem lain yang berbeda, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian *SMS Gateway* dapat diartikan sebagai suatu penghubung untuk lalu lintas data-data SMS, baik yang dikirim maupun yang diterima.

Pada awalnya, *SMS Gateway* dibutuhkan untuk menjembatani antara SMSC. Hal ini dikarenakan SMSC yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri yang bersifat pribadi. Sebagai contoh, Nokia memiliki protokol SMSC yang disebut dengan CIMD, sedangkan CMG memiliki protokol yang disebut dengan EMI. *SMS Gateway* ini kemudian ditempatkan diantara kedua SMSC berbeda tersebut, yang berfungsi sebagai relay bagi keduanya, yang kemudian akan menerjemahkan data dari protokol SMSC satu ke protokol SMSC lainnya yang dituju. Namun seiring dengan perkembangan teknologi komputer, SMS gateway lebih diartikan sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (*handphone*) dengan perangkat komputer, yang menjadikan aktifitas SMS menjadi lebih mudah dan menyenangkan (Yunianto, 2006).

2.2.6 Aplikasi Mobile

Aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain. (Buyens, 2010)

Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile* menurut adalah sebutan untuk aplikasi yang berjalan di *mobile device*. Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam

aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya.

Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemai oleh pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita mejadi semakin mudah menikmati hiburan kapan saja dan dimanapun. 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. (Purnama, 2010).

2.3 Alat Bantu Dalam Analisa dan Perancangan Sistem

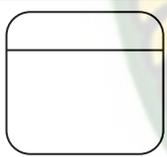
2.3.1 Data Flow Diagram (DFD)

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 terstruktur (*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.

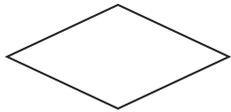
2.3.2 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga

menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi.

ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data (*database*). Model data ini juga akan membantu pada saat melakukan analisis dan perancangan basis data, karena model data ini akan menunjukkan bermacam-macam data yang dibutuhkan dan berhubungan antar data. ERD ini juga merupakan model konseptual yang dapat mendeskripsikan hubungan antara file yang digunakan untuk memodelkan struktur data serta hubungan antar data. (Yakub, 2012).

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. Adapun simbol kardinalitas dalam *Entity Relationship Diagram* (ERD) dapat dilihat pada tabel 2.3 berikut ini.

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

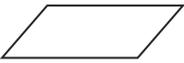
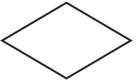
Flowchart adalah bagan yang menggambarkan urutan instruksi proses dan hubungan suatu proses dengan proses lainnya menggunakan simbol-simbol tertentu. Bagan alir digunakan sebagai alat bantu komunikasi dan dokumentasi. Pada saat akan menggambarkan suatu bagan alir analisis sistem atau programmer dapat mengikuti pedoman-pedoman sebagai berikut

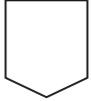
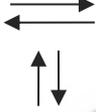
1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan 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 (Yakub, 2012).

Tabel 2.4 Simbol *Flowchart*

Simbol	Arti
	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 JQuery Mobile

JQuery merupakan suatu *framework (library)* Javascript yang menekankan bagaimana interaksi antara Javascript dan HTML. *JQuery* pertama kali dirilis pada tahun 2006 oleh John Resig. Fitur utama dari JQuery diantaranya :

1. Dapat mengakses elemen dalam dokumen

Javascript khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan Document Object Model dan pengaksesan harus secara spesifik menyesuaikan dengan struktur HTML.

2. Mengubah tampilan halaman website

CSS (Cascading Style Sheet) menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman web.

3. Mengubah isi dari dokumen

Tidak hanya memberikan “kosmetik” pada halaman web, JQuery juga memberikan fasilitas untuk mengubah isi dari dokumen hanya dengan beberapa baris perintah.

4. Merespon interaksi user

Javascript sendiri memiliki beberapa event-handling seperti onclick untuk menangani event saat terjadi click.

5. Animasi pada dokumen

Animasi seringkali disertakan dalam suatu halaman web untuk menambah kecantikannya.

6. Mengambil informasi dari server tanpa harus me-refresh halaman

Mengambil informasi dari server tanpa refresh halaman merupakan salah satu konsep dasar yang dikenal dengan nama AJAX (Asynchronous Javascript and XML).

7. Menyederhanakan penulisan sintaks Javascript

Semboyan JQuery adalah “Write less, do more” atau dengan kata lain kesederhanaan dalam penulisan code, tetapi menghasilkan tampilan yang lebih. (Wahana Komputer, 2012)

2.4.2 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang bersifat ringan. Sangat mudah bagi pengguna /developer untuk membaca dan menulis. Sangat mudah untuk mesin mengurai dan membuatnya. Hal ini didasarkan pada subset bahasa pemrograman JavaScript, Standar ECMA-262 Edisi 3 - Desember 1999. JSON merupakan format teks yang benar-benar bahasa yang independen tetapi menggunakan konvensi yang akrab bagi programmer dari bahasa keluarga-C , termasuk C, C ++, C #, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini membuat JSON menjadi mekanisme data-interchange

(Pertukaran data) antar bahasa pemrograman yang ideal. JSON dibangun pada dua struktur:

1. Kumpulan pasangan nama/nilai. Dalam berbagai bahasa, hal ini direalisasikan sebagai objek, record, struct, kamus, hash table, daftar ketik, atau array asosiatif.
2. Sebuah daftar urutan nilai. Dalam kebanyakan bahasa perograman, hal ini direalisasikan sebagai sebuah array, vektor, list, atau urutan.

JSON bersifat universal struktur data. Hampir semua bahasa pemrogramanmodern mendukung mereka dalam satu bentuk atau bentuk lain. In menyatakan bahwa format data yang dipertukarkan dengan bahasa pemrograman juga didasarkan pada struktur ini. Dalam JSON, mereka mengambil bentuk sebuah objek yang merupakan sebuah set urutan tak urut dari pasangan nama / nilai. Sebuah objek dimulai dengan '{' (kurung kurawa kiri) dan diakhiri dengan '}' (kurung kurawa kanan). Setiap nama diikuti oleh ':' (titik dua) dan nama / pasangan nilai dipisahkan dengan ',' (koma).

2.4.3 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. (Nugroho, 2012).

2.4.4 HTML

Hypertext Markup Language (HTML) adalah salah satu format yang digunakan untuk menulis halaman *web*, HTML ini berjalan di *web browser* dan memiliki fungsi untuk melakukan pemrograman aplikasi di atas *web*. HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu standar *Generalized Markup Language*.

Sebuah file HTML merupakan sebuah file teks yang berisi tag-tag *markup*. Tag *markup* memberitahukan *browser* bagaimana harus menampilkan sebuah halaman. File pada HTML harus memiliki ekstensi *htm* atau *html*. Selain itu file HTML dapat dibuat dengan menggunakan editor teks yang biasa dipakai. (Prasetio, 2014)

2.4.5 PHP

PHP merupakan singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan.

PHP memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa-bahasa sejenisnya, yaitu:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan *apache*, dengan *configurasi* yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem. (Anhar, 2010).

2.4.6 CSS

CSS (Cascading Style Sheet) merupakan bahasa *scripting* yang sangat ringan dan mudah dipelajari. CSS biasanya digunakan untuk menentukan tema tampilan dari sebuah website. Dengan menggunakan CSS, dapat ditentukan tampilan suatu elemen yang sama pada setiap halaman sehingga tidak perlu dilakukan penulisan kode program secara berulang-ulang untuk menampilkan tampilan yang sama (Nugroho, 2012).

2.4.7 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. Bersifat *open source*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
 2. MySQL merupakan sebuah *database server*, jadi dengan menggunakan database ini dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
 3. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
 4. Sistem *software*-nya tidak memberatkan kerja dari *server* atau komputer, karena dapat bekerja di *background*.
 5. MySQL didukung oleh driver ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
 6. Cukup aman karena memiliki *password* untuk mengaksesnya
 7. Mendukung field yang dijadikan sebagai kunci primer dan kunci unik.
- (Prasetio, 2014).