

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Sejarah**

Universitas Komputer Indonesia (UNIKOM) secara resmi berdiri pada hari Selasa, tanggal 8 Agustus 2000 berdasarkan Surat Keputusan Menteri Pendidikan Nasional nomor 126/D/0/2000.

Awalnya dimulai pada bulan Juli tahun 1994 ketika didirikan Lembaga Pendidikan Komputer Indonesia Jerman, disingkat LPKIG, bertempat di jalan Dipati Ukur 102 Bandung . Dengan 1 ruang kelas berkapasitas 50 orang dan 1 laboratorium komputer dengan 25 unit komputer, Lembaga ini membuka program pendidikan 1 tahun dengan 5 program studi yaitu Ahli Komputer Aplikasi Bisnis, Ahli Komputer Keuangan & Perbankan, Ahli Komputer Akuntansi & Perpajakan, Ahli Komputer Manajemen & Pemasaran dan Sekretaris Eksekutif. Jumlah peserta pendidikan pada tahun pertama ini sebanyak 233 siswa.

Pada tahun kedua, 1995, dibuka jenjang pendidikan 3 tahun untuk memenuhi animo siswa tahun pertama yang ingin memperdalam ilmunya, disamping pemikiran jangka panjang pengembangan institusi. Pada tahun ini juga dibuka program studi baru, meliputi : Ahli Komputer Teknik Informatika, Ahli Komputer Manajemen Informatika dan Sekretaris Eksekutif. Ruang kelas ditambah menjadi 2 buah dan laboratorium komputer menjadi 2 buah dengan jumlah siswa sebanyak 457 orang.

Pada tahun ketiga, 1996, dilakukan penambahan gedung kuliah baru bertempat di jalan Dipati Ukur 116 (gedung FISIP sekarang), sekaligus pemindahan pusat administrasi dan perkantoran. Dikedung baru ini dilakukan penambahan 1(satu) Lab. Komputer, 5(lima) Ruang Kuliah, Ruang Dosen dan Ruang Kemahasiswaan. Jumlah siswa dari tahun 1996 hingga tahun 1998 bertambah dari 632 orang menjadi 1184 orang.

Pada tahun kelima, 1998, dimulai pembangunan Kampus baru (Gedung Rektorat /Kampus-1 sekarang) berlantai 6(enam) di jalan Dipati Ukur 114.

Pembangunan Kampus baru ini dapat diselesaikan pada bulan Agustus 1999, sehingga pada awal perkuliahan bulan September 1999 telah dapat digunakan.

Mencermati dinamika peserta didik dan pengembangan Institusi kedepan, pada tanggal 24 Desember 1998 dibentuklah Yayasan *Science* dan Teknologi dan dilanjutkan dengan pengajuan pendirian STIMIK IGI dan STIE IGI ke DIKTI.

Pada bulan Juli 1999 STIE IGI diresmikan dengan keluarnya SK Mendiknas no. 119/D/O/1999 dengan 5 program studi : Akuntansi S1, Manajemen S1, Manajemen Pemasaran D3, Keuangan Perbankan D3 serta Akuntansi D3.

Pada bulan Agustus 1999 STIMIK IGI diresmikan dengan keluarnya SK Mendiknas no. 143/D/O/1999 dengan 5 program studi : Teknik Informatika S1, Manajemen Informatika D3, Teknik Komputer D3, Komputerisasi Akuntansi D3 serta Teknik Informatika D3.

Agar Sistem Pendidikan lebih Efisien, Efektif, Produktif dengan Struktur Organisasi yang lebih baik, enam bulan kemudian dilakukan usulan ke DIKTI untuk melakukan *Merger* kedua Sekolah Tinggi diatas menjadi Universitas.

Pada hari Selasa, tgl. 8 Agustus 2000 keluarlah SK MENDIKNAS no. 126/D/O/2000 atas Universitas Komputer Indonesia yang disingkat dengan nama UNIKOM. Pada SK tersebut sekaligus diijinkan dibukanya 11 program studi baru : Teknik Komputer S1, Manajemen Informatika S1, Teknik Industri S1, Teknik Arsitektur S1, Perencanaan Wilayah dan Kota S1, Ilmu Hukum S1, Ilmu Komunikasi S1, Ilmu Pemerintahan S1, Desain Interior D3, Desain Komunikasi Visual S1 dan Desain Komunikasi Visual D3.

Sejak berdirinya pada tahun 2000, setiap tahunnya UNIKOM menerima ± 2.000 mahasiswa baru. Terakhir pada tahun 2014 yang lalu diterima sebanyak 3.108 mahasiswa baru. Hingga tahun akademik 2015/2016 terdapat 6 Fakultas dan 23 Program Studi di UNIKOM dengan jumlah mahasiswa sebanyak 15.000 orang yang berasal dari berbagai pelosok tanah air dan dari luar negeri yang sedang menempuh pendidikan di UNIKOM.

### **2.1.1. Visi, Misi, dan Tujuan**

Berikut adalah visi, misi, dan tujuan dari Universitas Komputer Indonesia (UNIKOM):

#### **2.1.1.1. Visi**

Menjadi Universitas terkemuka dibidang Teknologi Informasi & Komunikasi, berwawasan Global, berjiwa *Entrepreneur* dan menjadi Pusat Unggulan dibidang Ilmu Pengetahuan dan Teknologi yang mendukung Pembangunan Nasional serta berorientasi pada kepentingan Masyarakat, Bangsa dan Negara.

#### **2.1.1.2. Misi**

Menyelenggarakan Pendidikan Tinggi Modern berdasarkan Budaya Organisasi UNIKOM, PIQIE (*Professionalism, Integrity, Quality, Information Technology, Excellence*), dengan Sistem Pendidikan yang Kondusif dan Program-program Studi yang berbasis pada Software (Perangkat Lunak), *Hardware* (Perangkat Keras), dan *Entrepreneurship* (Kewirausahaan) dengan mengoptimalkan Sumber Daya yang ada berdasarkan prinsip Efisiensi, Efektifitas dan Produktifitas.

#### **2.1.1.3. Tujuan**

Menghasilkan Lulusan yang unggul dibidang Teknologi Informasi & Komunikasi, Kompeten dan Handal di Bidang Studinya, berjiwa *Entrepreneur*, Santun dan Berbudi Luhur, Memiliki Komitmen untuk memajukan Bangsa dan Negara serta Beriman dan Bertaqwa kepada Tuhan Yang Maha Esa.

### 2.1.2. Logo

Berikut adalah logo dari Universitas Komputer Indonesia (UNIKOM) yang dapat dilihat pada Gambar 2.1:



sumber : <https://www.unikom.ac.id>

**Gambar 2. 1 Logo Universitas Komputer Indonesia (UNIKOM)**

### 2.2. Aplikasi

Aplikasi dalam konteks ini adalah aplikasi perangkat lunak adalah sebuah program atau perangkat lunak yang ditujukan untuk tujuan tertentu yang dapat membantu pengguna dalam menyelesaikan atau mempercepat penyelesaian masalah dengan memanfaatkan kemampuan komputer. Menurut buku Pengantar Basis Data oleh Sri Widiyanti (2000), aplikasi adalah sebuah perangkat lunak yang menjadi *front end* dalam sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang berguna untuk orang-orang dan sistem yang bersangkutan [4]. Contoh aplikasi dapat membantu mempercepat sebuah pekerjaan yang biasa dilakukan manusia adalah aplikasi pengolah kata, sebelum munculnya sebuah aplikasi perangkat lunak pengolah kata, manusia menggunakan sebuah perangkat mesin ketik yang di mana apabila melakukan sebuah kesalahan maka harus mengulang proses pengetikan, dengan aplikasi pengolah kata hal tersebut dengan mudah cukup dengan menghapus bagian yang perlu diubah. Contoh lain adalah aplikasi pembuatan tagihan berbasis *web*, aplikasi ini berguna untuk membuat tagihan secara otomatis setiap bulan yang dikirimkan melalui email, tanpa perlu seseorang membuat tagihan kepada sekumpulan pelanggan yang dikirim melalui surat.

### 2.2.1. Aplikasi Berdasarkan Fungsi

Aplikasi perangkat lunak dapat dibagi menjadi beberapa macam berdasarkan fungsinya, yaitu:

1. *Word Processing*

*Word Processing* adalah aplikasi perangkat lunak yang berfungsi untuk melakukan pengolahan kata seperti menulis dokumen atau surat-menyurat.

Contoh: Microsoft Word, Google Docs

2. *Desktop Publishing*

*Desktop Publishing* adalah aplikasi perangkat lunak yang berfungsi untuk membantu dalam penulisan naskah.

Conth: Final Draft, Celtx, Trelby

3. *Program Speadsheet*

Program Speadsheet adalah aplikasi perangkat lunak yang berfungsi untuk melakukan perhitungan matematika dengan layout baris dan kolom.

Contoh: Microsoft Excel, Google Sheets

4. *Database Management System*

*Database Management System* adalah aplikasi perangkat lunak yang berfungsi untuk melakukan pengolah basis data. Pengolahan ini dapat berupa penambahan, pengubahan, pembacaan, dan penghapusan data.

Contoh: Microsoft Access, phpMyAdmin

5. *Graphics*

*Graphics* adalah aplikasi perangkat lunak yang berfungsi untuk melakukan pengolahan sebuah citra.

Contoh: Adobe Photoshop, Paint, Corel Draw

6. *Program Akuntansi*

Aplikasi yang berfungsi untuk pengolahan keuangan dan akuntansi.

Contoh: Zoho Books, ZipBooks

7. *Program Statistik*

Aplikasi perangkat lunak yang berfungsi untuk pengolahan data yang mampu menghasilkan informasi yang berguna.

Contoh: SPSS, Stata, GraphPad

8. *Communication*

Aplikasi perangkat lunak yang berfungsi untuk membantu dalam hubungan komunikasi jarak jauh antar dua atau lebih orang dalam waktu yang bersamaan atau berbeda.

Contoh: Skype, WhatsApp, Gmail

9. *Multimedia*

Aplikasi multimedia adalah aplikasi perangkat lunak yang berfungsi untuk melakukan pekerjaan multimedia yang mencakup beberapa aspek seperti suara, video, teks, dan gambar.

Contoh: Adobe Premiere Pro, Blender

10. *Game*

Aplikasi *game* adalah aplikasi perangkat lunak yang berfungsi untuk melakukan kegiatan permainan yang menghibur dengan media video.

Contoh: Pinball, Solitaire

11. *Anti virus*

Aplikasi anti virus adalah aplikasi yang berfungsi untuk mendeteksi virus yang terdapat pada komputer dan membantu membersihkannya.

Contoh: AVG, McAfee, Smadav

**2.3. Text to Speech**

Sistem Text to Speech pada prinsipnya terdiri dari dua sub sistem [5], yaitu:



**Gambar 2. 2 Skema Text To Speech**

## 1. Bagian Konverter Teks ke Fonem (Text to Phoneme)

Bagian Konverter Teks ke Fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta pitch-nya. Bagian ini bersifat sangat language dependant. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut.

## 2. Bagian Konverter Fonem ke Ucapan (Phoneme to Speech)

Bagian Konverter Fonem ke Ucapan menerima masukan berupa kode-kode fonem serta pitch dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah format *synthesizer* serta *diphone concatenation*. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. *Formant synthesizer* bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk menghasilkan sinyal ucapan yang diinginkan. *Synthesizer* jenis ini telah lama digunakan pada berbagai aplikasi. Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, *synthesizer* ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

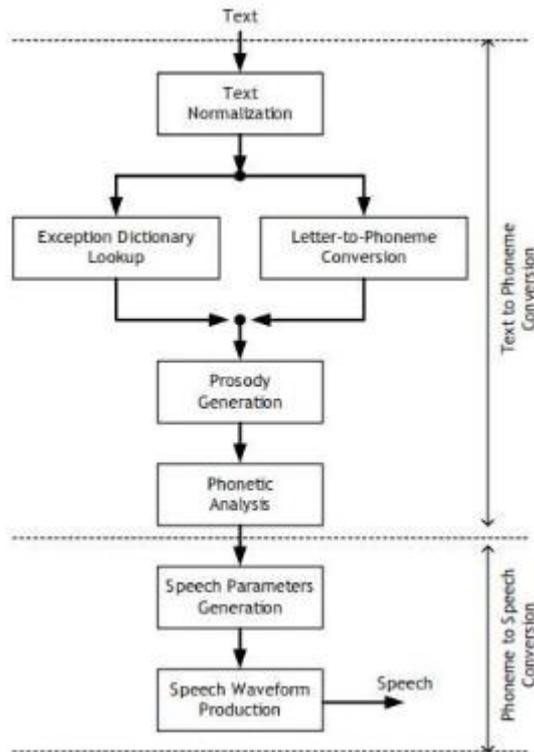
*Synthesizer* yang menggunakan teknik *diphone concatenation* bekerja dengan cara menggabung-gabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa diphone (gabungan dua buah fonem). *Synthesizer* jenis ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian (*naturalness*) yang tinggi.

Struktur sistem seperti di atas pada prinsipnya merupakan konfigurasi tipikal yang digunakan pada berbagai sistem Text to Speech berbagai bahasa. Namun demikian, pada setiap sub-sistem terdapat sifat-sifat serta proses-proses yang sangat spesifik dan sangat tergantung dari bahasanya.

Konversi dari teks ke fonem sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol fonetik yang merepresentasikan unit bunyi terkecil dalam suatu bahasa. Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini menyebabkan implementasi unit konverter teks ke fonem

menjadi sangat spesifik terhadap suatu bahasa. Untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (prosody). Secara kuantisasi, prosodi adalah perubahan nilai pitch (frekuensi dasar) selama pengucapan kalimat dilakukan atau pitch sebagai fungsi waktu. Pada prakteknya, informasi pembentuk prosodi berupa data-data pitch serta durasi pengucapannya untuk setiap fonem yang dibangkitkan. Nilai-nilai yang dihasilkan diperoleh dari suatu model prosodi. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Beberapa model umum prosodi pernah dikembangkan, tetapi untuk digunakan pada suatu bahasa masih perlu banyak penyesuaian yang harus dilakukan.

Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Pada sistem yang menggunakan teknik diphone concatenation, sistem harus didukung oleh suatu diphone database yang berisi rekaman segmen-segmen ucapan yang berupa diphone. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda. Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan dengan diagram seperti terlihat pada gambar 2.3.



sumber : <http://eprints.undip.ac.id/25758/1/ML2F002573.pdf>

**Gambar 2. 3 Urutan Proses Konversi dari Teks ke Ucapan**

#### 2.4. API (*Application Programming Interface*)

API (*Application Programming Interface*) merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan *software* yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari *software* tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut.

## 2.5. Javascript

Menurut Yeni Kustiyahningsih dan Devie Rosa Anamisa, 2011 JavaScript adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap HTML dengan mengizinkan pengeksekusian perintah-perintah disisi user artinya disisi browser bukan disisi server web. JavaScript adalah bahasa yang “case sensitive” artinya menandakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama TEST berbeda dengan variabel dengan nama test dan setiap instruksi diakhiri dengan karakter titik koma.

### 2.5.1. Sejarah Javascript

Awal tahun 1995, Netscape baru saja meluncurkan dukungan terhadap Java Applet dalam browser. Untuk mempermudah programmer non-Java dalam mengakses Java Applet tersebut, maka Netscape mengembangkan Javascript. Ditujukan untuk desainer dan pengembang web, Javascript dirancang untuk dapat digunakan tanpa memerlukan pengetahuan yang terlalu mendalam tentang pemrograman berorientasi objek.

Di masa sekarang, 2014 dan selanjutnya, Javascript telah menjadi lingua franca dari dunia web dalam browser. Meskipun telah lahir bahasa-bahasa lain untuk pengembangan web, bahasa-bahasa tersebut biasanya dikompilasi ke Javascript. Javascript juga telah merambah ke dunia server, dan digunakan di luar konteks browser dan HTML. Javascript tak lagi terikat dengan HTML dan *browser*, dan tak lagi hanya digunakan untuk program-program kecil [6].

## 2.6. JSON

*JSON* adalah sebuah format pesan balikan yang berukuran kecil yang mudah dibaca dan ditulis oleh manusia, dan juga mudah ditulis dan diurai oleh mesin [7]. *JSON* merupakan salah satu pesan balikan yang dapat digunakan dalam sebuah *REST API*. Format lain yang sering digunakan adalah *XML*.

*JSON* terbagi dalam dua struktur, yaitu [7]:

1. Gabungan *name/value*, atau yang biasa dikenal *object* atau *record* dalam bahasa pemrograman.

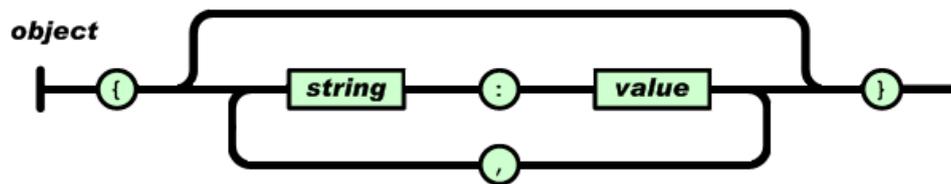
2. *List value*, atau yang biasa dikenal larik dalam bahasa pemrograman.

### 2.6.1. Format Bentuk JSON

JSON memiliki format-format tipe data berikut yang dapat digunakan [7]: objek, larik, *value*, *string*, dan angka. Berikut adalah penjelasan dari masing-masing bentukan format tersebut:

#### 2.6.1.1. object

Berikut adalah format *JSON object* yang dapat dilihat pada Gambar 2.4:



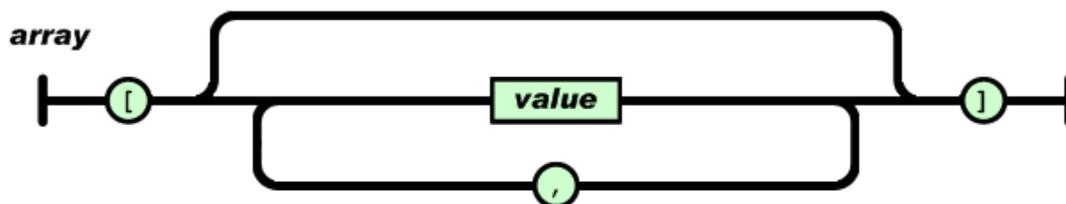
Sumber gambar: <http://json.org>

**Gambar 2. 4 Format JSON Object**

*Object* adalah sekumpulan dari pengenal dan nilai. Sebuah *object* diawali dengan karakter “{” (kurung kurawal kiri) dan diakhiri dengan karakter “}” (kurung kurawal kanan). Masing-masing pengenal diikuti dengan karakter “:” (titik dua) dan masing-masing nilai diakhiri dengan karakter “,” (koma).

#### 2.6.1.2. array

Berikut adalah format *JSON array* yang dapat dilihat pada Gambar 2.5:



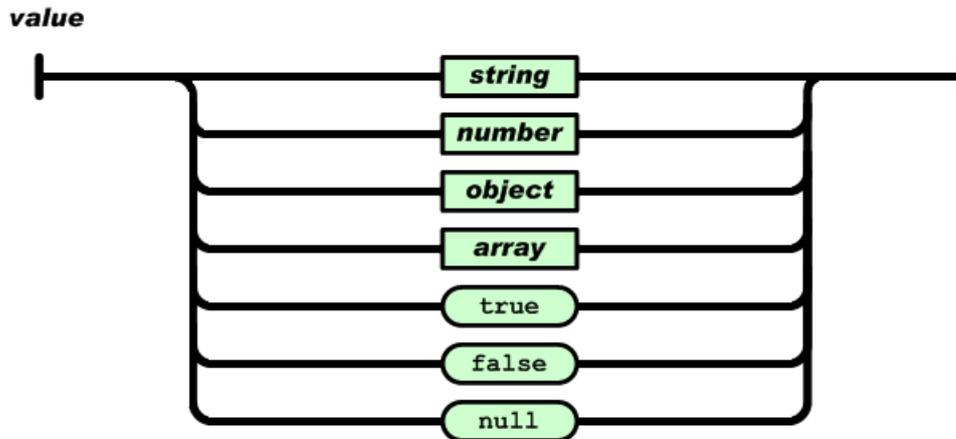
Sumber gambar: <http://json.org>

**Gambar 2. 5 Format JSON Array**

*Array* adalah sekumpulan nilai. Sebuah *array* diawali dengan karakter “[” (kurung siku kiri) dan diakhiri dengan karakter “]” (kurung siku kanan). Masing-masing nilai dipisahkan dengan karakter “,” (koma).

#### 2.6.1.3. value

Berikut adalah format *JSON value* yang dapat dilihat pada Gambar 2.6:



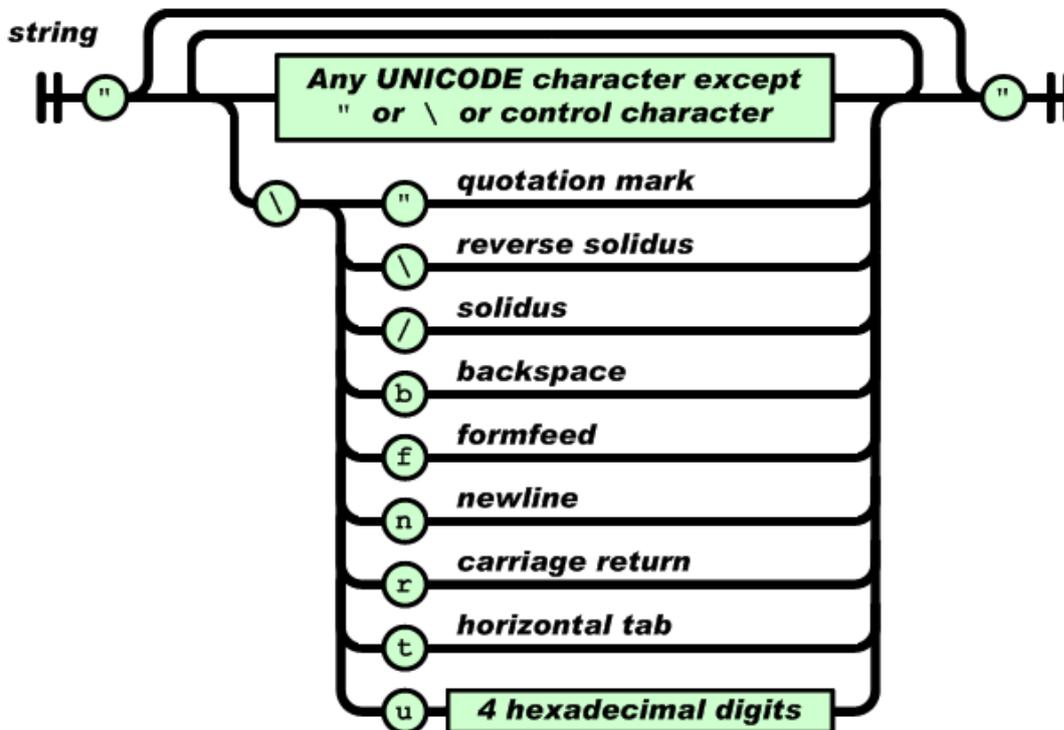
Sumber gambar: <http://json.org>

**Gambar 2. 6** Format JSON *value*

Sebuah *value* dapat bernilai *string* diapit dengan karakter petik dua, angka, *boolean* atau *null*, *object*, atau *array*.

#### 2.6.1.4. String

Berikut adalah format *JSON string* yang dapat dilihat pada Gambar 2.7:



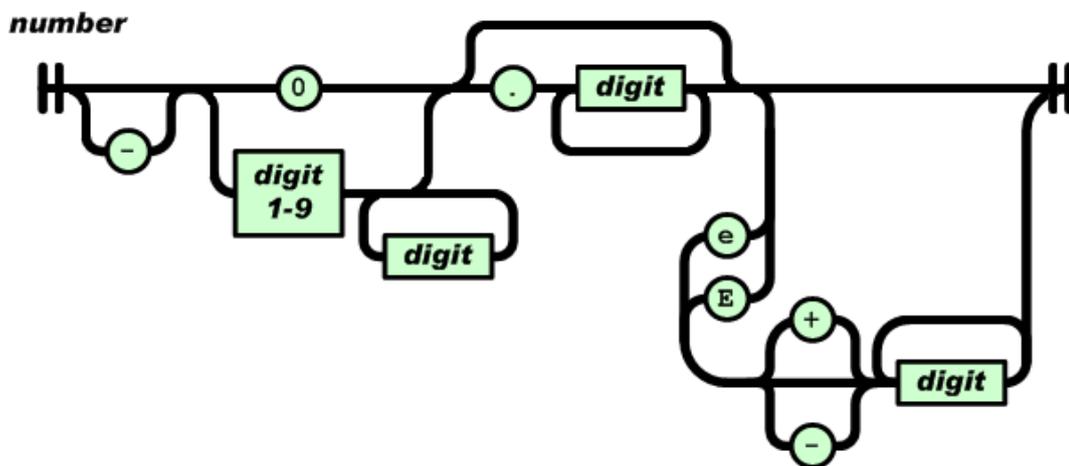
Sumber gambar: <http://json.org>

**Gambar 2. 7** Format JSON *String*

*String* adalah sekumpulan dari karakter kosong atau karakter Unicode yang diapit oleh karakter petik dua atau karakter khusus diawali dengan karakter “\” (garis miring terbalik). Sebuah karakter dianggap sebagai sebuah *string* dengan panjang satu karakter. *String* mirip dengan tipe data *string* pada bahasa pemrograman C atau Java.

#### 2.6.1.5. number

Berikut adalah format *JSON number* yang dapat dilihat pada Gambar 2.8:



Sumber gambar: <http://json.org>

**Gambar 2. 8 Format JSON Number**

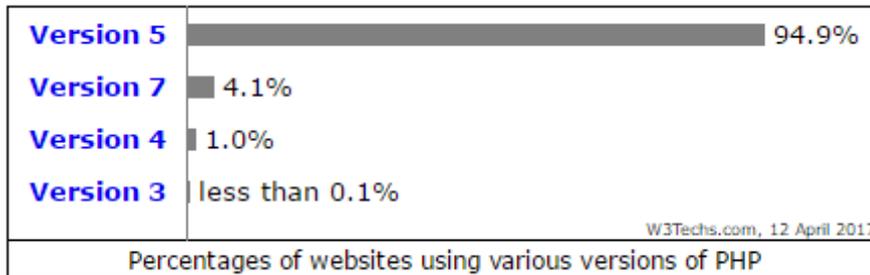
*Number* memiliki kemiripan dengan tipe data *number* pada bahasa pemrograman C atau Java, namun yang membedakan adalah pada *string JSON* tidak menggunakan format oktal dan hexadesimal.

## 2.7. PHP

PHP adalah sebuah bahasa pemrograman sumber terbuka yang ditujukan untuk pemrograman *web* dan dapat diaplikasikan ke *HTML*. *PHP* merupakan bahasa skrip yang ditanam dalam *HTML*. Ini berarti bahwa Anda dapat menggabungkan kode *PHP* dan *HTML* dalam berkas yang sama [8]. *PHP* merupakan bahasa pemrograman *server-side* yang di mana program akan dijalankan di *server* dan hasilnya akan diintegrasikan ke dalam kode sumber *HTML*. *PHP* diciptakan untuk membuat halaman *web* yang interaktif.

Dengan tujuan utama untuk membuat halaman *web* yang interaktif tersebut, *PHP* cukup dikenal oleh banyak kalangan pemrogram *web*. Menurut W3Techs—

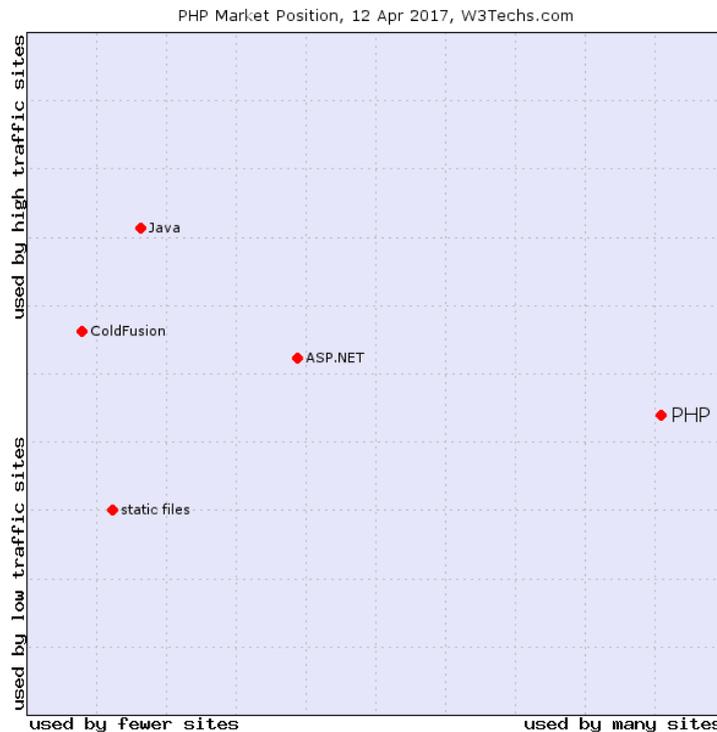
lembaga survei teknologi *web*—menyatakan 82.7% *website* yang menggunakan pemrograman *server-side*, menggunakan bahasa pemrograman PHP [9]. Versi PHP yang paling digunakan saat ini adalah PHP versi 5 dan diikuti dengan versi 7. Berikut adalah perbandingan penggunaan versi PHP yang dapat dilihat pada gambar 2.9:



Sumber: <http://w3techs.com>

**Gambar 2. 9 Perbandingan penggunaan versi PHP**

Dan gambar berikut menyatakan perbandingan bahasa pemrograman PHP dengan bahasa pemrograman populer yang lain yang dapat dilihat pada gambar 2.10:



Sumber: <http://w3techs.com>

**Gambar 2. 10 Perbandingan PHP dengan bahasa pemrograman lain**

PHP juga memiliki fitur yang cukup mumpuni untuk mengakomodasi dalam kebutuhan basis data. Terdapat modul untuk menggunakan basis data yang otomatis terinstal pada saat instalasi PHP pada komputer berhasil. Basis data yang didukung oleh PHP cukup beragam, misal, MySQL, Oracle DB, PostgreSQL, MongoDB, SQLite, dan lain-lain.

Program PHP diapit dengan sintak `<?php ?>` dan ekstensi format *file* php adalah `.php`. Program PHP dapat digabung dengan sintak HTML, namun hanya akan membaca kode yang berada di antara sintak `<?php ?>`, hasil pemrosesan akan ditampilkan kembali ke dalam kode HTML. Contoh sebuah program PHP:

```

1. <!doctype html>
2. <html>
3. <head>
4.     <title>Contoh Program PHP dalam HTML</title>
5. </head>
6. <body>
7.     <p>
8.         <?php
9.             echo 'Sekarang adalah tahun ' . date('Y') . '.';
10.        ?>
11.     </p>
12. </body>
13. </html>

```

Program di atas, setelah diproses oleh kompiler PHP akan menghasilkan:

```

1. <!doctype html>
2. <html>
3. <head>
4.     <title>Contoh Program PHP dalam HTML</title>
5. </head>
6. <body>
7.     <p>
8.         Sekarang adalah tahun 2017.
9.     </p>
10. </body>
11. </html>

```

## 2.8. MySQL

MySQL adalah sebuah aplikasi *server* basis data yang dijalankan di *server*, MySQL menggunakan sintaks SQL standar dalam penggunaannya. MySQL bersifat relasional yang memungkinkan satu tabel dapat berelasi dengan tabel lainnya, sehingga memungkinkan dilakukannya normalisasi untuk mencegah redundansi dalam pemakaian data. MySQL mampu menampung data yang sangat

besar. MySQL menggunakan InnoDB mampu menampung sebanyak 1017 kolom dan besar data yang mencapai 256TB [10].

MySQL memiliki tipe-tipe bidang data yang dapat dituliskan ke dalam tabelnya. Tipe-tipe tersebut adalah format yang dapat diterima untuk setiap isiannya. Beberapa tipe bidang akan dijelaskan pada Tabel 2.1 [8].

**Tabel 2. 1 Tipe-tipe bidang pada MySQL**

Tipe Bidang	Penjelasan Singkat
<i>char</i>	Sembarang data berbasis karakter dapat disimpan pada bidang ini, tetapi bidang tersebut akan memiliki panjang yang dibatasi dengan nilai di dalam kurung.
<i>varchar</i>	Sembarang data berbasis karakter dapat ditempatkan pada bidang ini, dan data tersebut dapat memiliki panjang yang bervariasi dari 0 sampai 255 karakter. Panjang maksimum bidang diberikan pada nilai di dalam kurung.
<i>int</i>	<i>Integer</i> yang memiliki rentang dari -2,147,483,648 sampai +2,147,483,647 dapat disimpan pada bidang ini. Parameter <i>length</i> membatasi jumlah digit yang dapat ditampilkan, bukan nilainya.
<i>text</i>	Sembarang data berbasis karakter dapat ditempatkan pada bidang ini, dengan ukuran maksimum sebesar 65,536 karakter.
<i>decimal</i>	Bidang numerik yang dapat menyimpan nilai desimal. Parameter <i>length</i> membatasi jumlah digit yang ditampilkan, dan parameter <i>dec</i> membatasi jumlah digit di belakang titik desimal yang dapat disimpan.
<i>enum</i>	Dipakai hanya untuk nilai-nilai tertentu, seperti <i>"true"</i> dan <i>"false"</i> .
<i>datetime</i>	Bidang multiguna yang dapat menyimpan tanggal dan waktu secara bersamaan sebagai YYYY-MM-DD' hh:mm:ss.

Berikut adalah beberapa contoh sintak dasar yang perlu diketahui dalam mengelola sebuah basis data MySQL:

### 1. Membuat *Database*

```
1. create table db1;
```

### 2. Membuat Tabel

```
1. create table mahasiswa (
2. mahasiswa_id int(6) auto_increment primary key,
3. nim varchar(12),
4. nama_awal varchar(30) not null,
5. nama_akhir varchar(30) not null,
6. date_added datetime default CURRENT_TIMESTAMP
7. )
```

### 3. Menambahkan Data

```
1. insert into mahasiswa (nim, nama_awal, nama_akhir)
2. values ('10113081', 'Iqbal', 'Perkasa');
```

#### 4. Mengambil Data

```
1. select * from mahasiswa;
```

#### 5. Mengubah Data

```
1. update mahasiswa set nama_akhir = 'Perdana' where nim = '10113081';
```

#### 6. Menghapus Data

```
1. delete from mahasiswa where nim = '10113081';
```

#### 7. Menghapus Tabel

```
1. drop table mahasiswa;
```

### 2.9. Basis Data

Basis data merupakan suatu kumpulan data yang terhubung dan disimpan secara bersamaan pada suatu media penyimpanan. Data disimpan dengan cara-cara tertentu sehingga dapat dengan mudah untuk digunakan atau ditampilkan kembali, dapat digunakan oleh satu atau lebih sistem secara optimal. Data juga disimpan tanpa mengalami ketergantungan dengan program atau aplikasi yang akan menggunakannya, data juga disimpan agar pada saat proses penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan lebih terkontrol dengan baik [11].

Basis data adalah kumpulan data yang saling berelasi. Data sendiri, merupakan fakta mengenai obyek, orang dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol) [12].

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut [12]:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa *redundancy* yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/label/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

## 2.10. Entity Relationship Diagram (ERD)

Entity Relationship Diagram ERD adalah pemodelan awal basis data yang paling banyak digunakan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen yang dikembangkan oleh Peter Chen, notasi Barker yang dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis, notasi Crow's Foot dan beberapa notasi lain. Namun pada umumnya banyak digunakan adalah notasi Chen. Model ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau.

ERD mempunyai notasi-notasi diantaranya adalah sebagai berikut:

### 1. Entiti

Entiti adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai atau pelaku dalam sistme yang akan dibuat.

### 2. Atribut

Entiti mempunyai elemen yang disebut atribut dan berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja.

### 3. Hubungan

Menunjukkan adanya hubungan / relasi diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Hubungan antar entitas memiliki derajat untuk menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas yang lain, berikut adalah jenis-jenis derajat pada ERD:

#### a. Satu ke satu (one to one)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling dengan satu entitas pada himpunan entitas A.

b. Satu ke banyak (one to many)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

c. Banyak ke satu (many to one)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan B.

d. Banyak ke banyak (many to many)

Setiap entitas pada himpunan A dapat berhubungan banyak dengan banyak entitas pada himpunan entitas B demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A [13].

### 2.11. Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

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 atau dimana data tersebut disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

- a. Kebebasan dari menjalankan implementasi teknis sistem.
- b. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
- c. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.

- d. Menganalisis sistem yang diajukan untuk menentukan apakah data data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

- a. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
- b. Membedakan sistem dari lingkungannya dengan menempatkan batas batasnya.
- c. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
- d. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

DFD terdiri dari context diagram dan diagram rinci (DFD Levelled). Context diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam DFD levelled akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam DFD levelled bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level  $x$  harus berhubungan dengan aliran data yang masuk dan keluar pada level  $x+1$  yang mendefinisikan proses pada level  $x$  tersebut. Proses yang tidak dapat diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif [14].

## 2.12. Kamus Data

Merupakan katalog (tempat penyimpanan) dari elemen-elemen yang berada dalam satu sistem. Kamus data mempunyai fungsi yang sama dalam pemodelan sistem dan juga berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail, dan me-reorganisasi semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

- a. Menjelaskan arti aliran data dan penyimpanan dalam DFD.
- b. Mendeskripsikan komposisi paket data yang bergerak melalui aliran misalnya alamat diuraikan menjadi kota, negara dan kode pos.
- c. Mendeskripsikan komposisi penyimpanan data.
- d. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran data.
- e. Mendeskripsikan hubungan detail antar penyimpanan yang akan menjadi titik perhatian dalam Diagram Keterhubungan Entitas (E-R) [14].

