



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

Berikut ini merupakan teori umum yang dibutuhkan di dalam laporan:

2.1.1 Pengertian Perangkat Lunak (*Software*)

“Perangkat lunak komputer adalah perangkat berbasis program (tidak berbentuk fisik) atau algoritma yang digunakan untuk mengoperasikan dan mengontrol perangkat keras komputer”. (Fauzi, 2018)

Sedangkan, menurut Utami dan Asnawati (2015:2), “Perangkat lunak (*software*) adalah program yang berisi kumpulan instruksi untuk melakukan proses pengolahan data”.

Berdasarkan beberapa definisi perangkat lunak diatas, dapat disimpulkan bahwa perangkat lunak adalah instruksi-instruksi atau data yang diformat secara digital, yang bisa dibaca dan ditulis oleh komputer sesuai kehendak pemakai.

2.1.2 Pengertian Basis Data (*Database*)

Menurut Rosa dan Shalahuddin (2015:43) “Basis data merupakan salah satu bagian dalam rekayasa perangkat lunak yang terkomputerisasi dan bertujuan utama memelihara data yang sudah diolah atau media penyimpanan informasi agar dapat diakses dengan mudah dan cepat.”

Sedangkan, menurut Subandi (2018:3), “Basis data dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat.”

Berdasarkan pendapat diatas, dapat disimpulkan bahwa basis data (*database*) kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan suatu program komputer agar dapat memudahkan mendapatkan informasi.



2.2 Teori Khusus

Dalam teori ini meliputi penjelasan tentang metode pengembangan sistem menggunakan metode waterfall, black-box testing beserta tahapan tahapannya, menjelaskan tentang kamus data, diagram konteks, data flow diagram (DFD), block chart, flow chart dan juga tentang penjelasan entity relationship diagram (ERD).

2.2.1 Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan adalah metode waterfall. Pressman dikutip Sanubari et all (2020:41-42) menyatakan bahwa metode air terjun atau sering disebut metode waterfall sering dinamakan siklus hidup klasik (*classic life cycle*) dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak. Dalam pengembangan metode waterfall memiliki beberapa tahapan yang berurut yaitu : *Requirement Analysis* (Analisis Kebutuhan), *Design System* (Desain Sistem), *Implementation* (Implementasi), *Integrating & Testing* (Integrasi & Pengujian), *Operation & Maintance* (Penerapan & Pemeliharaan Program) tahapan - tahapan dari metode waterfall adalah sebagai berikut :

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan Batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi, survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. Sistem Desain (*System Design*)

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.



3. Implementasi (*Implementation*)

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi pada tahap selanjutnya, setiap unit dikembangkan dan di uji fungsionalitas yang disebut sebagai unit testing.

4. Integrasi & Pengujian (*Integrating & Testing*)

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem di uji untuk mengecek setiap kegagalan maupun kesalahan.

5. Penerapan & Pemeliharaan Program (*Operation & Maintance*)

Tahap akhir dalam metode waterfall. Perangkat lunak yang sudah jadi,dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada Langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2.2 *Black-Box Testing*

Hadiprakoso (2020:285) menyatakan bahwa, “dalam pengujian *Black-Box*, anda melihat perangkat lunak sebagai kotak hitam yang tidak bisa anda lihat isi (kode) di dalamnya. Anda tahu apa yang seharusnya dilakukan perangkat lunak tersebut tetapi anda tidak tahu cara kerjanya. Sekalipun tidak mengetahui cara kerjanya masih dapat melakukan pengujian dengan memberikan sebagai macam input kemudian melihat outputnya seperti apa. Pengujian *Black-Box*, juga disebut pengujian perilaku atau pengujian fungsional, karena berfokus pada fungsi perangkat lunak. Meskipun anda tidak tahu cara kerja perangkat lunak, anda dapat mencoba menebak nilai yang mungkin membuat perangkat lunak gagal berfungsi.”

Menurut Saputra et al (2020:170) “pengujian dengan menggunakan *Black-Box* di gunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang telah di rancang”.



2.2.3 Kamus Data

Kadir (2016:73) menyatakan bahwa, “Kamus data adalah deskripsi formal mengenai seluruh elemen yang tercakup dalam DAD pada tahapan perancangan elemen-elemen pada kamus data akan menjadi bahan untuk menyusun basis data”. Berikut beberapa simbol-simbol yang terdapat pada kamus data :

Tabel 2.1 Simbol-simbol Kamus Data

Simbol	Artinya
=	Terdiri dari, mendefinisikan, diuraikan menjadi
+	Dan
()	Menunjukkan suatu elemen yang bersifat pilihan(opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file
[]	Menunjukkan salah satu dari dua situasi tertentu satu elemn bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada searah bersamaan. Elemen-elemen yang ada didalam tanda kurung ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)
**	Komentar
	Pemisah sejumlah alternatif pilihan antara simbol []
@	Identifikasi atribut kunci



{ }	<p>Menunjukkan elemen-elemen repetitive, juga disebut kelompok berulang atau tabal-tabel. Kemungkinan bisa ada satu atau beberapa elemen berluang didalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan</p>
-----	---

Sumber : Maniah dan Hamidin (2017:60)

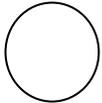
2.2.4 Diagram Konteks

“Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari Data Flow Diagram yang menggambarkan seluruh input ke sistem atau output dari sistem”. (Oktafianto, 2016:45)

Menurut Mukhtar “Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran dari sistem”. (2018:82)

Kesimpulannya, Diagram konteks adalah level diagram tertinggi dari *Data Flow Diagram* yang terdiri dari suatu proses dan menggambarkan antara *entity* luar, masukan, dan keluaran dari suatu sistem. Pembuatan diagram konteks memiliki 3 simbol utama :

**Tabel 2.2** Simbol-simbol dalam Diagram Konteks

Simbol	Nama Simbol	Keterangan
	Entitas Exsternal	Terminator menggambarkan asal atau tujuan data dari luar sistem.
	Proses	Menggambarkan entitas atau proses dimana aliran data, data masukan di transformasikan ke aliran data keluar
	Arus data	Menggambarkan aliran data.

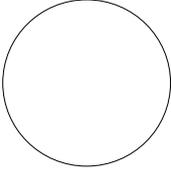
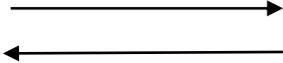
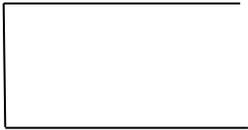
Sumber : Hidayat (2018:13)

2.2.5 Data Flow Diagram (DFD)

Menurut Kurniawan (2020:11) “*Data Flow Diagram* adalah diagram arus data yang menggunakan symbol-simbol untuk menyajikan entitas, proses, arus data, dan penyimpanan yang berkaitan dengan suatu sistem dengan tujuan untuk memberikan indikasi mengenai bagaimana data di transformasi pada saat data bergerak melalui sistem dan menggambarkan fungsi-fungsi (dan sub fungsi) yang mentransformasikan aliran data”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa DFD adalah diagram yang menggunakan notasi-notasi dalam perancangan terstruktur, bertujuan untuk menggambarkan aliran data pada sistem yang berjalan secara logika. Adapun beberapa simbol yang biasanya digunakan dalam perancangan DFD, diantaranya seperti pada tabel dibawah ini:

**Tabel 2.3** Simbol-simbol *Data Flow Diagram* (DFD)

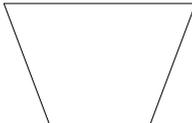
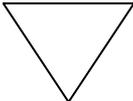
Notasi	Keterangan
	Proses, Simbol ini digunakan untuk proses transformasi data
	<i>External Entity</i> , Simbol ini digunakan untuk menggambarkan asal atau tujuan data.
	Data <i>flow</i> symbol ini digunakan untuk menggambarkan aliran data yang berjalan
	Data <i>store</i> , simbol ini digunakan untuk menggambarkan data flow yang sudah disimpan atau diarsipkan

Sumber : Sutabri (2012:116)

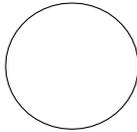
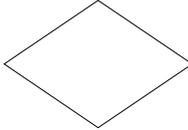
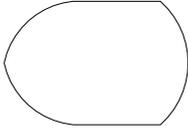
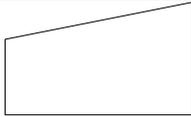
2.2.6 Block Chart

Menurut Rusmawan (2019:75), *Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Tabel 2.4 Simbol *Block Chart*

Simbol	Arti
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
	Multi dokumen
	Proses manual
	Proses yang dilakukan oleh komputer
	Menandakan dokumen yang diarsipkan (arsip manual)
	Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain



	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
	Terminasi yang menandakan awal dan akhir dari suatu aliran
	Pengambilan keputusan (<i>decision</i>)
	Layar peraga (<i>monitor</i>)
	Pemasukan data secara manual

Sumber: Rusmawan (2019:75-77)

2.2.7 Flow Chart

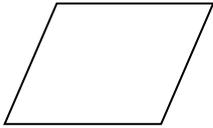
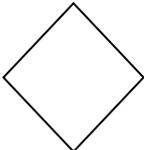
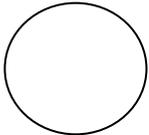
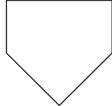
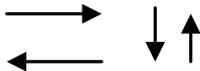
“*Flow Chart* adalah algoritma dalam suatu program yang menyatakan arah alur program tersebut.” (Qoiriah et all, 2019:12)

“*Flow Chart* adalah diagram yang menunjukkan alir di dalam program atau prosedur sistem secara logika.” (Nurmalina et all, 2017 : 277). Berikut simbol-simbol yang ada pada *Flow Chart* :

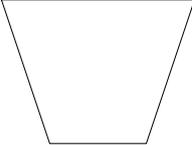
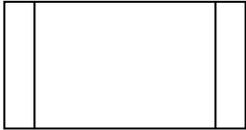
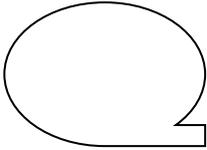
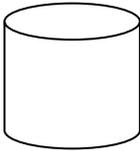
Tabel 2.5 Simbol-simbol *Flow Chart*

Simbol <i>Flow Chart</i>	Fungsi <i>Flow Chart</i>
	Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flow chart</i> .



	<p>Simbol pemrosesan yang terjadi pada sebuah alur kerja.</p>
	<p>Simbol input atau output yang mendefinisikan masukan dan keluaran proses.</p>
	<p>Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.</p>
	<p>Simbol konektor untuk menyambung proses pada lembar kerja yang sama.</p>
	<p>Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda</p>
	<p>Simbol untuk menghubungkan antar proses atau antar symbol.</p>
	<p>Simbol yang menyatakan piranti keluaran seperti layar monitor, printer dan lain-lain.</p>



	<p>Simbol yang mendefinisikan proses yang dilakukan secara manual.</p>
	<p>Simbol masukan atau keluaran dari atau ke sebuah dokumen.</p>
	<p>Simbol yang menyatakan bagian dari program (sub-program).</p>
	<p>Simbol masukan atau keluaran dari atau ke sebuah pita magnetik.</p>
	<p>Simbol database atau basis data.</p>

Sumber : Rusmawan (2019:49)

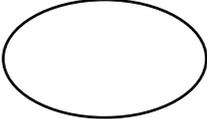
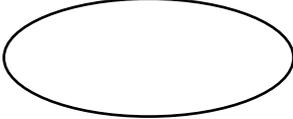
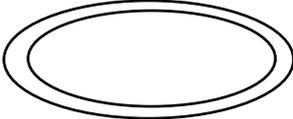
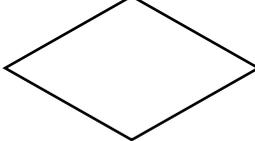
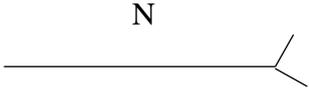
2.2.8 Entity Relationship Diagram (ERD)

Menurut Sukamto dan Shalahuddin (2016:50), “ERD digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.”



Simbol-simbol *Entity Relationship Diagram* (ERD) dalam notasi Chen dapat dilihat pada tabel di bawah ini.

Tabel 2.6 *Entity Relationship Diagram* (ERD)

Simbol	Deskripsi
	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.
	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.
	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki kemungkinan jumlah pemakaian. Jumlah maksimum keterhubungan antara entitas satu dengan yang lain disebut dengan kardinalitas.

Sumber: Sukamto dan Shalahuddin (2016:50-51)



2.3 Teori Khusus

Agar penyusunan Laporan Akhir ini lebih terarah dan tidak menyimpang dari permasalahan yang ada, maka penulis membatasi ruang lingkup permasalahan yang akan dibahas diantaranya:

1. Sistem Informasi Akademik dalam lembaga ini memiliki 4 aktor, yaitu peserta didik, admin, pengajar, dan pimpinan.
2. Peserta didik dapat melakukan registrasi akun, melakukan pendaftaran dan pembayaran bimbingan belajar, melihat rapor, dan mendapatkan sertifikat.
3. Admin dapat mengelola data tingkatan dan jadwal pembelajaran, mengelola data pengajar dan jenis pembelajaran, melihat data pendaftaran dan pembayaran bimbingan belajar peserta didik.
4. Pengajar dapat mengelola data nilai peserta didik.
5. Pimpinan dapat melihat laporan pendaftaran dan pembayaran peserta didik, serta laporan hasil akhir peserta didik.
6. Sistem informasi yang dibangun menggunakan bahasa pemrograman *PHP* dan *MySQL*.

2.3.1 Pengertian Sistem

Menurut Romney & Steinbart (dalam Mulyani, 2017) mengatakan sistem adalah kumpulan dari dua atau lebih komponen yang saling bekerja dan berhubungan untuk mencapai tujuan.

2.3.2 Pengertian Informasi

“Definisi kata ‘informasi’ sendiri secara internasional telah disepakati sebagai hasil dari pengolahan data yang secara prinsip memiliki nilai atau *value* yang lebih dibandingkan dengan data mentah”. (Rukun & Hayadi, 2018:2)



2.3.3 Pengertian Sistem Informasi

Menurut Turban, McLean, dan Wetherbe (dalam Rukun & Hayadi, 2018:1), sistem informasi adalah sebuah sistem yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik”.

2.3.4 Pengertian Sistem Informasi Akademik

“Sistem informasi akademik sangat bermanfaat dan berperan aktif dalam institusi pendidikan, hal ini tercermin dari penerapan sistem informasi dalam proses pembelajaran, jadwal kelas, penyusunan jadwal ujian, yang merupakan tugas bagian pengajaran”. (Simarmata et all, 2020:50)

Sedangkan, menurut Vindi dan Sabandi (dalam Simarmata et all, 2020:51), sistem informasi akademik adalah segala macam hasil interaksi antara elemen di lingkungan akademi untuk menghasilkan informasi yang kemudian dijadikan landasan pengambilan keputusan, melaksanakan tindakan, baik pelaku proses itu sendiri, maupun dari pihak luar”.

2.3.5 Sistem Informasi Akademik Bahasa Inggris berbasis *Website* pada Lembaga Pendidikan *Intensive English Course Palembang*

Sistem Informasi Akademik Bahasa Inggris berbasis *Website* pada Lembaga Pendidikan *Intensive English Course Palembang* adalah sistem informasi yang digunakan oleh Lembaga Pendidikan *Intensive English Course Palembang* untuk mengelola jadwal yang dilakukan oleh admin dan mengelola nilai peserta bimbingan belajar yang dilakukan oleh pengajar. Data tersebut akan diteruskan ke peserta bimbingan belajar. Nilai akhir yang diterima oleh peserta bimbingan belajar berupa sertifikat.



2.4 Teori Program

2.4.1 Pengertian XAMPP



Gambar 2.1 Logo XAMPP

“XAMPP adalah perangkat lunak sumber terbuka yang dikembangkan oleh teman-teman Apache. Paket perangkat lunak XAMPP berisi distribusi Apache untuk server Apache, MariaDB, PHP, dan Perl” (Aprilian & Saputra, 2020:3).

Menurut Wicaksono (2019:138), “XAMPP merupakan kompilasi dari beberapa aplikasi gratis yang mendukung banyak sistem operasi. XAMPP merupakan aplikasi dalam platform, Apache, MySQL, PHP dan Perl. Fungsinya adalah sebagai server yang berdiri sendiri (localhost)”.

Berikut ini penjelasan singkat mengenai komponen-komponen yang ada di XAMPP :

1. *FileZilla*, digunakan untuk server pertukaran file melalui protokol *FTP*.
2. *Mercury Mail Server*, digunakan untuk server email sederhana.
3. *Tomcat*, aktifkan jika ingin mengaktifkan web server untuk *Java Servlet*, *Java Server Page (JSP)*, *Java EL* dan *WebSocket*.
4. *Apache*, adalah paket aplikasi yang digunakan untuk web server yang handal dan stabil.

2.4.2 Pengertian PHP



Gambar 2.2 Logo PHP



Menurut Sunoto (2015:149), “*PHP* adalah biasanya script yang menyatu dengan tag-tag *HTML* yang dieksekusi di server dan dikembangkan untuk pembuatan web dinamis.”

Disamping itu, menurut Betha (2017:31) *PHP* merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen *HTML* secara *on the fly* yang di server web, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML*.”

Contoh penulisan *script PHP* adalah sebagai berikut:

Tabel 2.7 *Script Dasar PHP*

<?	<div style="border: 1px solid black; width: 150px; height: 20px; margin: 0 auto; text-align: center;">.....</div>	Tempat	penulisan
?>			
Atau			
<?php	<div style="border: 1px solid black; width: 150px; height: 20px; margin: 0 auto; text-align: center;">.....</div>	Tempat	penulisan
?>			

Di dalam *script PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah *script*. Beberapa cara untuk memberi keterangan di dalam *script PHP* adalah sebagai berikut:

1. Gunakan tag */** dan akhiri tag **/* apabila jumlah keterangan lebih dari 1 baris.
2. Gunakan tag *//*, tag ini digunakan untuk keterangan yang hanya terdiri dari 1 baris saja.
3. Gunakan tag *#*, tag ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah *script PHP* selalu diakhiri dengan menuliskan tanda titik koma (;). Satu baris *script PHP* tidak harus berada dalam satu baris, melainkan dapat menuliskan perintah *script PHP* lebih dari satu baris.



2.4.3 MySQL

Bunafit (2019:134) menyatakan bahwa “MySQL merupakan *database* yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data.”

Sedangkan, menurut Subagia (2018:67), MySQL merupakan *software database open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL.”

Kesimpulannya, MySQL adalah *database server* yang gratis dengan lisensi *General Public License* (GPL) tanpa membayar lisensi yang ada.

2.4.4 Sekilas tentang Notepad ++



Gambar 2.3 Logo Notepad++

Menurut Habibi dan Suryansah (2020:63), “Notepad++ adalah suatu *text editor* yang berjalan pada *Operating System* (OS) Windows”.

Notepad++ menggunakan komponen-komponen *Scintilla* agar dapat menampilkan dan menyunting teks dan berkas *source code* berbagai bahasa pemrograman.

Keunggulan Notepad++ dalam *Software Web Programming* :

1. **Simple**, ringan dan cepat dibandingkan dengan *text editor* lainnya, karena tidak perlu menunggu *loading opening library* terutama pada PC/Laptop yang memiliki spesifikasi yang rendah.
2. **Bracket Matching**, biasanya digunakan pada saat menuliskan *syntax* percabangan, perulangan, dan bagian utama program.



3. **Syntax Highlighting**, tampilan *source code* dapat dilihat dengan warna yang berbeda pada setiap fungsi.
4. **Syntax Folding**, digunakan untuk menyembunyikan *block program* tertentu agar terlihat lebih ringkas tampilannya.
5. **Quick Color Picker++**, berguna pada saat kita menuliskan kode warna pada *HTML* ataupun *CSS*.
6. **FingerText**, biasa digunakan untuk menuliskan bahasa *PHP* di *Notepad++*, berfungsi untuk memudahkan pengetikan *syntac* dengan kata tertentu sebagai pemicu dan menggantikannya dengan menekan tombol *Tab*.