**BAB II**

**TINJAUAN PUSTAKA**

Pada bab kedua ini akan dibahas mengenai beberapa definisi yang berkaitan dengan judul tugas akhir dan juga teori serta istilah dalam penyusunan tugas akhir yang dimaksud. Dengan adanya pembahasan dalam bab kedua ini diharapkan tugas akhir yang disusun dapat lebih mudah dipahami.

**2.1 Pengertian Pembangunan Aplikasi**

**2.1.1 Pengertian Pembangunan**

Menurut Kamus Besar Bahasa Indonesia Edisi Ketiga (2002:103) pembangunan adalah proses, cara, pebuatan membangun.

Pembangunan adalah sebagai suatu proses, akan terkait dengan mekanisme sistem atau kinerja suatu sistem. Menurut Easton (dalam Miriam Budiardjo, 1985), proses sistemik paling tidak terdiri atas tiga unsur: pertama, adanya input, yaitu bahan masukan konversi, kedua, adanya proses konversi, yaitu wahana untuk ”mengolah” bahan masukan, ketiga, adanya output, yaitu sebagai hasil dari proses konversi yang dilaksanakan. Proses sistemik dari suatu sistem akan saling terkait dengan subsistem dan sistem-sistem (mathedu-unila.blogspot.com/2011/12).

**2.1.2 Pengertian Aplikasi**

Sebagaimana yang tertera dalam [www.ipulhe.com/pengertian aplikasi](http://www.ipulhe.com/pengertian%20aplikasi): 2015 bahwa aplikasi dapat dikatakan suatu perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari user atau pengguna, aplikasi banyak diciptakan guna membantu berbagai keperluan seperti untuk laporan, percetakan dan lain-lain. Sedangkan istilah aplikasi berasal dari istilah bahasa inggris *application*  yang berarti penerapan, lamaran atau penggunaan. Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut.

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

**2.1.3 Pembangunan Aplikasi**

Pembangunan aplikasi adalah suatu proses pembuatan program komputer yang ditulis dalam suatu bahasa pemrograman yang dirancang untuk menjalankan perintah-perintah dari pengguna untuk mencapai tujuan tertentu.

**2.2 Pengertian Pengolahan Format Data Borang Akreditasi**

**2.2.1 Pengertian Pengolahan**

Pengolahan berasal dari kata dasar [olah](https://www.apaarti.com/olah.html). Pengolahan memiliki arti dalam kelas [nomina](https://www.apaarti.com/nomina.html) atau kata benda sehingga pengolahan dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan.

Pengolahan menurut Tim Penyusun Kamus Pusat Pembinaan dan Pengembangan Bahasa (1988) adalah sebuah proses mengusahakan atau mengerjakan sesuatu supaya menjadi lebih sempurna.

* + 1. **Pengertian Format Data**

Format menurut Tim Penyusun Kamus Pusat Bahasa (2008) adalah bentuk dan ukuran.

Menurut Turban (2010:41) data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan tetapi belum terorganisir untuk menyampaikan arti tertentu.

Format data adalah bentuk deskripsi dari suatu benda, peristiwa, aktivitas dan transaksi yang dikelompokkan dan diorganisir untuk menyampaikan suatu informasi.

* + 1. **Pengertian Borang Akreditasi**

Borang adalah alat untuk mengumpulkan dan mengungkapkan data dan informasi yang digunakan untuk menilai kelayakan dan mutu sebuah instituisi.

Akreditasi adalah proses evaluasi dan penilaian mutu institusi atau program studi yang dilakukan oleh suatu tim pakar sejawat (tim asesor) berdasarkan standar mutu yang telah ditetapkan, atas pengarahan suatu badan atau lembaga akreditasi mandiri diluar institusi atau program studi yang bersangkutan. Hasil akreditasi merupakan pengakuan bahwa suatu institusi atau program studi telah memnuhi standar mutu yang telah ditetapkan itu, sehingga layak untuk menyelenggarakan program-programnya .

Status akreditasi suatu perguruan tinggi merupakan cermin kinerja perguruan tinggi yang bersangkutan dan menggambarkan mutu, efisiensi, serta relevansi suatu program studi yang diselenggarakan. Saat ini terdapat dua jenis akreditasi yang diberikan oleh pemerintah kepada program studi di perguruan tinggi, yaitu:

1. Status Terdaftar, Diakui, atau Disamakan yang diberikan kepada Perguruan Tinggi Swasta.
2. Status Terakreditasi atau Nir-Akreditasi yang diberikan kepada semua perguruan tinggi (Perguruan Tinggi Negeri, Perguruan Tinggi Swasta, dan Perguruan Tinggi Kedinasan).

Karena adanya dua status akreditasi yang sama-sama masih berlaku, saat ini terdapat PTS yang menyandang kedua-duanya untuk program studinya. Hal ini terjadi karena proses pemberian status akreditasi dilakukan melalui dua jalur yang berbeda sesudah terbentuknya Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Sebelumnya, penentuan status didasarkan pada Surat Edaran Dirjen Dikti No.470/D/T/1996.

Kemudian pemerintah menetapkan, untuk pelaksanaan akreditasi terhadap suatu PTS/Unit PTS, sepanjang belum pernah dievaluasi (diakreditasi) oleh atau melalui BAN-PT, akan tetap dilakukan berdasarkan peraturan tersebut diatas, tetapi manakala suatu PTS/Unit PTS telah pernah dievaluasi (diakreditasi) oleh atau melalui BAN-PT, maka selanjutnya pelaksanaan akreditasi terhadap PTS yang bersangkutan dilakukan dengan berpedoman pada kriteria atau Borang Akreditasi dari BAN-PT.

Pada bulan Desember 1994 dibentuk BAN-PT untuk membantu pemerintah dalam upaya melakukan tugas dan kewajiban melaksanakan pengawasan mutu dan efisiensi pendidikan tinggi. Pembentukan BAN-PT ini menunjukkan bahwa akreditasi perguruan tinggi di Indonesia pada dasarnya adalah tanggungjawab pemerintah dan berlaku bagi semua perguruan tinggi, baik negeri maupun swasta. Hal ini sekaligus menunjukkan niat dan kepedulian pemerintah dalam pembinaan penyelenggaraan perguruan tinggi, melayani kepentingan masyarakat, dan kemajuan ilmu pengetahuan dan teknologi untuk meningkatkan taraf kehidupan masyarakat dan memperkaya kebudayaan nasional.

Karena tidak lagi membedakan negeri dan swasta, pengertian akreditasi dalam dunia pendidikan tinggi adalah pengakuan atas suatu lembaga pendidikan yang menjamin standar minimal sehingga lulusannya memenuhi kualifikasi untuk melanjutkan pendidikan ke tingkat yang lebih tinggi atau memasuki pendidikan spesialisasi, atau untuk dapat menjalankan praktek profesinya.

Akreditasi perguruan tinggi yang diterapkan dalam sistem pendidikan nasional dimaksudkan untuk menilai penyelenggaraan pendidikan tinggi. Penilaian itu diarahkan pada tujuan ganda, yaitu:

1. Menginformasikan kinerja perguruan tinggi kepada masyarakat.
2. Mengemukakan langkah pembinaan yang perlu ditempuh terutama oleh perguruan tinggi dan pemerintah, serta partisipasi masyarakat.

Peringkat pengakuan yang diberikan oleh pemerintah pada perguruan tinggi didasarkan atas hasil akreditasi perguruan tinggi yang dilaksanakan oleh BAN-PT, dengan melakukan akreditasi yang meliputi akreditasi lembaga dan akreditasi program studi.

Kriteria penilaian untuk akreditasi lembaga terdiri dari :

1. Izin penyelenggaraan pendidikan tinggi.
2. Persyaratan dan kelayakan penyelengaraan pendidikan tinggi.
3. Relevansi penyelenggaraan program pendidikan dengan pembangunan.
4. Kinerja perguruan tinggi.
5. Efisiensi pengelolaan perguruan tinggi.

Kriteria penilaian untuk akreditasi program studi terdiri atas :

1. Identitas
2. Izin penyelenggaraan program studi.
3. Kesesuaian penyelenggaraan program studi dengan peraturan perundang-undangan.
4. Relevansi penyelenggaraan program studi.
5. Sarana dan prasarana.
6. Efisiensi penyelenggaraan program studi.
7. Produktivitas program studi.
8. Mutu lulusan.

Klasifikasi penilaian untuk semua kriteria tersebut ditentukan oleh 3 aspek, yaitu mutu (bobot 50%), efisiensi (25%), dan relevansi (25%). Sesudah melalui penghitungan semua nilai kriteria, didapat peringkat akreditasi perguruan tinggi sebagai berikut:

|  |  |
| --- | --- |
| Nilai | Peringkat |
| 0-400 | NA |
| 401-500 | C |
| 501-600 | B |
| 601-700 | A |

**Tabel 2.1** Nilai dan Peringkat Akreditasi Perguruan Tinggi

Mengingat jumlah perguruan tinggi yang menjadi sasaran saat ini lebih dari 1400, serta bentuk dan ragam program pendidikan yang diselenggarakan, akreditasi perguruan tinggi dilaksanakan secara bertahap dan berkesinambungan. Pelaksanaan akreditasi perguruan tinggi oleh BAN-PT diawali dengan melakukan uji coba pada beberapa perguruan tinggi yang menyelenggarakan satu program studi. Selanjutnya dilaksanakan secara berkala dan bertahap serta terus menerus.

Standar akreditasi adalah tolak ukur yang harus dipenuhi oleh institusi ataupun program studi. Standar akreditasi terdiri atas beberapa parameter yang dapat digunakan sebagai dasar:

1. Penyajian data dan informasi mengenai kinerja, keadaan dan perangkat kependidikan program studi, yang dituangkan dalam instrumen akreditasi.
2. Evaluasi dan penilaian mutu kinerja, keadaan dan perangkat kependidikan program studi.
3. Penetapan kelayakan program studi untuk menyelenggarakan program-programnya.
4. Perumusan rekomendasi perbaikan dan pembinaan mutu program studi.

Standar akreditasi program studi mencakup standar tentang komitmen program studi terhadap kapasitas institusional (institusional capacity) dan komitmen terhadap efektivitas program pendidikan (educational effectiveness), yang dikemas dalam tujuh standar akreditasi, yaitu:

|  |  |
| --- | --- |
| Standar 1. | Visi, misi tujuan dan sasaran, serta strategi pencapaian. |
| Standar 2. | Tata pamong, kepemimpinan, sistem pengelolaan, dan penjaminan mutu. |
| Standar 3. | Mahasiswa dan lulusan. |
| Standar 4. | Sumber daya manusia. |
| Standar 5. | Kurikulum, pembelajaran, dan suasana informasi. |
| Standar 6. | Pembiayaan sarana dan prasarana, serta sistem informasi. |
| Standar 7. | Penelitian dan pelayanan/pengabdian kepada masyarakat dan kerjasama. |

* + 1. **Pengolahan Format Data Borang Akreditasi**

Pengolahan Format Data Borang Akreditasi adalah suatu proses untuk mendapatkan data yang sesuai dengan format data kuantitatif Borang Akreditasi.

**2.3 Basis Data**

Basis data (*database*) adalah sekumpulan dari berbagai data yang saling berhubungan satu dengan lainnya. Basis data tersimpan di perangkat keras, serta dimanupulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. Basis data merupakan salah satu komponen penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau *user*.

Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data atau disingkat DBMS (Database Management System). Manipulasi basis data meliputi pembuatan pernyataan (query) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (update) data, serta pembuatan report data.

Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi user. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisien yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas struktur data. Basis data menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan update menjadi rumit.

**2.4 Perangkat Desain**

**2.4.1 UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) adalah bahasa pemodelan yang menggunakan konsep orientasi objek. Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkungan sistem dan bagaimana hubungan antara sistem dengan subsitem maupun sistem lain diluarnya. UML dibuat oleh Grady Booch, James Rumbaugh dan Ivar Jacobson di bawah bendera *Rational Software Corp*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek.

1. *Use Case Diagram,* untuk memodelkan proses bisnis.
2. *Conceptual Diagram*, untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram*, untuk memodelkan pengiriman pesan antar *object*.
4. *Collaboration Diagram*, untuk memodelkan interakasi antar *object*.
5. *State Diagram*, untuk memodelkan perilaku *object* di dalam sistem.
6. *Activity Diagram*, untuk memodelkan perilaku *use case* dan *object* di dalam sistem.
7. *Class Diagram*, untuk memodelkan struktur kelas.
8. *Object Diagram*, untuk memodelkan struktur *object*.
9. *Component Diagram*, untuk memodelkan komponen *object*.

10. *Deployment Diagram*, untuk memodelkan distribusi aplikasi.

Dalam pembangunan sebuah aplikasi berbasis objek tidak semua diagram tersebut digunakan. Paul R. Reed, Jr. menyatakan bahwa dalam sebuah pembangunan aplikasi minimal terdiri dari *class* *diagram, sequence diagram*, dan *use case diagram*.

* + 1. **Use Case Diagram**

*Use case* diagram adalah suatu kumpulan urutan interaksi diantara *user* dengan sistem untuk mencapai suatu tujuan dimana *use case* ini menggambarkan kebutuhan fungsional suatu sistem tanpa menampilkan struktul internal sistem.

*Use case* diagram menjelaskan urutan kegiatan yang dilakukan actordan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* diagram ini hanya menjelaskan apa yang dilakukan oleh *actor* terhadap sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

Beberapa komponen kunci analisis *use case* diagram adalah :

1. *Actor*, entitas yang menggunakan atau yang digunakan sistem. Umumnya orang, namun bisa juga sistem eksternal.
2. *Connection*, penghubung antar actor ke *use case*.
3. *Relationship*, hubungan antara actor ke *use case*.

Beberapa simbol dalam *use case* diagram yaitu :

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Nama | Keterangan |
| 1 |  | *Actor* | Menunjukkan pelaku yang berinteraksi dengan sistem |
| 2 |  | *Association* | Komunikasi antara *actor* dan *use case* yang berpartisipasi pada *use case* |
| 3 |  | *Extend* | Relasi *use case* tambahan ke sebuah *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu |
| 4 |  | *Generalization* | Hubungan generalisasi dan spesialisasi antar dua buah *use case*. Menunjukkan objek induk yang berbagi perilaku dengan objek anak. |
| 5 |  | *Include* | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya. |
| 6 |  | *System Boundary* | Menunjukkan batas dari suatu sistem. |
| 7 |  | *Use Case* | Menunjukkan suatu fungsi yang bisa dilakukan. |

**Tabel 2.2** Simbol *Use Case* Diagram

(Sumber Modul *Workshop* UML Bab 2)

* + 1. **Sequence Diagram**

*Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu.

Diagram ini secara khusus berasosiasi dengan use case diagram. Memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case.

Berikut beberapa simbol yang terdapat di dalam sequence diagram:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Nama | Keterangan |
| 1 |  | *Actor* | Menunjukkan pelaku yang berinteraksi dengan sistem |
| 2 |  | *Object* | *Object* atau yang biasa disebut partisipan merupakan *instance* dari sebuah *class*. |
| 3 |  | *Lifeline* | Menunjukkan waktu mulai dan berakhirnya pesan. |
| 4 |  | *Activation* | *Activation* dinotasikan sebagai sebuah kotak segiempat yang digambar pada sebuah *lifeline*. *Activation* mengindikasikan sebuah objek yang akan melakukan sebuah aksi. |
| 5 |  | *Call Message* | *Call Message* menyatakan suatu objek atau memanggil operasi/*method* yang ada pada objek lain atau dirinya sendiri. |
| 6 |  | *Send Message* | *Send Message* menyatakanbahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah kepada objek yang dikirim |
| 7 |  | *Return Message* | *Return Message* menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau *method* menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 8 |  | *Create Message* | *Create Message* menyatakan suatu objek dapat membuat objek lain, arah panah mengarah pada objek yang dibuat. |
| 9 |  | *Destroy Message* | *Destroy Message* menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri |
| 10 |  | *Self-Message* | *Self-Message* menunjukkan komunikasi ke dalam sebuah objek itu sendiri. |

**Tabel 2.3** Simbol *Sequence* Diagram

(Sumber : Martin Fowler UML Distilled 2005, Sholiq Pemodelan Sistem Informasi Berorientasi Objek dengan UML 2006)

* + 1. **Activity Diagram**

*Activity* diagram menggambarkan berbagai alir aktivitas dalam sistem aplikasi bagaimana masing-masing alir berawal, decision(keputusan) yang mungkin terjadi, dan bagaimana mereka berakhir.

Pada UML digunakan untuk memperlihatkan urutan aktivitas proses bisnis. Dapat juga digunakan untuk menggambarkan aliran kejadian (*work flow event*) dalam *use case*.

Beberapa simbol dalam *activity* diagram :

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Nama | Keterangan |
| 1 |  | *Initial State* | Menunjukkan titik awal, untuk memulai suatu aktivitas. |
| 2 |  | *Final State* | Menunjukkan titik akhir, untuk mengakhiri suatu aktivitas. |
| 3 |  | *Activity* | Menunjukkan aktifitas yang berjalan, aktivitas yang dilakukan sistem. |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 |  | *Decision* | Menunjukkan percabangan (alur data berbeda sesuai kondisi yang berlaku). Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 5 |  | *Fork and Join* | Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
| 6 |  | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

**Tabel 2.4** Simbol *Activity* Diagram

(Sumber : *User Guide Enterprise Architect* 7.0)

* + 1. **Class Diagram**

*Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan method. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Beberapa simbol dalam class diagram adalah :

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Simbol | Nama | Keterangan |
| 1 |  | *Association* | Menunjukkan hubungan antar kelas yang biasanya hanya berupa hubungan parameter atau *return value*. Garis ini melambangkan tipe-tipe *relationship* dan juga menampilkan hukum-hukum multiplisitas pada sebuah *relationship* (seperti *one to one, one to many dan many to many*) |
| 2 |  | *Aggregation* | Menunjukkan hubungan kelas induk dan anak, kelas anak masih bisa berdiri sendiri. |
| 3 |  | *Composition* | Menunjukkan hubungan kelas induk dan anak, kelas anak tidak bisa berdiri sendiri |
| 4 |  | *Dependency* | Menunjukkan suatu kelas yang bergantung pada kelas lainnya, perubahan pada kelas yang berdiri sendiri dapat mempengaruhi kelas yang bergantung padanya. |
| 5 |  | *Realization* | Menunjukkan realisasi dari suatu kelas, digunakan untuk menunjukkan kelas implement dari suatu interface. |
| 6 |  | *Class* | *Class* adalah blok-blok pembangunan pada program berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah nama *class*, bagian tengah mendefinisikan atribut dan bagian akhir mendefinisikan *method-method* dari sebuah *class*. |

**Tabel 2.5** Simbol *Class* Diagram

(Sumber : Modul *Workshop* UML Bab 2)

**2.5 Perangkat Coding**

* + 1. **Sublime Text 3**

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-*packages*.

Sublime Text bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML.

Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan *add-ons* yang bisa di *download* sesuai kebutuhan *user*.

Sublime Text ini memiliki banyak kelebihan diantaranya :

1. **Multiple Selection**

Mempunyai fungsi untuk melakukan perubahan pada sebuah kode dalam waktu yang sama dan dalam baris yang berbeda.

1. **Command Pallete**

Mempunyai fungsi yang berguna untuk mengakses file shortcut dengan mudah,untuk mencari file tersebut dengan menekan CTRL+SHIFT+P.

1. **Find free mode**

Fitur ini sangat dibutuhkan oleh pengguna yang sedang fokus dalam pekerjaan,yaitu dapat merubah tampilan penuh dengan tampilan SHIFT+F11.

1. **Find in project**

Kita dapat mencari *project* dan memilih file dalam *project*. Hanya dengan menekan Ctrl + P anda dapat mencari file yang diingikan.

1. **Plugin API Switch**

Sublime text mempunyai keunggulan dengan *plugin* yang berbasis Python Plugin API. Teks editor ini juga mempunyai *plugin* yang sangat beragam , dan ini dapat memudahkan pengguna dalam mengembangkan *software* nya.

1. **Drag & Drop**

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dala editor yang akan membuka tab baru secara otomatis.

1. **Split Editing**

Di dalam fitur ini pengguna dapat mengedit file secara berdampingan dengan klik File > New menu into file.

1. **Multi Platform**

Sublime text juga mempunyai keunggulan dalam bagian platform. Sublime text sendiri sudah tersedia dalam berbagai platform sistem operasi, yaitu Windows, Linux, dan MacOS.

* + 1. **PHP**

PHP (*Personal Home Page*) diciptakan oleh Ramus Lerdorf, seorang pemogram C yang andal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada homepagenya. Rasmus adalah seorang pendukung *open source*. Karena itulah ia mengeluarkan *Personal Home Page* Tools versi 1.0 secara gratis. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan PHP 1.0 dan menerbitkan PHP 2.0.

Menurut Bunafit Nugroho (2005:369) PHP adalah program aplikasi yang bersifat *server side*, artinya hanya dapat berjalan pada sisi *server* saja dan tidak dapat berfungsi tanpa adanya sebuah *server* di dalamnya. Secara khusus, PHP dirancang untuk membentuk web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Pada 43 prinsipnya, PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Serever Page*), Cold Fusion, maupun Perl.

PHP juga bukan bahasa pemograman yang lengkap. Maksudnya program ini tidak menyertakan *compiler* tersendiri yang membuat program hasilnya menjadi program .exe yang dapat di jalankan tersendiri. PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa pemograman *scripting* berlisensi *Open Source*. Script ini dapat bercampur dengan script Tag HTML sehingga karena kemampuannya tersebut, ia disebut bahasa yang *embeded* pada Tag HTML.

* + 1. **CodeIgniter**

CodeIgniter adalah sebuah framework php *open source* dengan menggunkan konsep MVC (*Model, View, Controller*) untuk membangun *website* dinamis atau aplikasi berbasis web. CodeIgniter ini dikembangkan oleh Rick Ellis yang di rilis pertama kali pada tanggal 28 Februari 2006.

Framework adalah kumpulan intruksi-intruksi yang di kumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan developer dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar tercipta nya *source code* yang bersih dan terstruktur.

Model MVC merupakan konsep yang cukup populer dalam pembangunan aplikasi web. MVC (*Model, View, Controller*) itu memisahkan antara logika pembuatan kode dengan pembuatan template website/tampilan dari web. Jika kita menggunakan Model-View-Controller (MVC) menjadikan pembuatan sebuah *website* akan menjadi lebih terstruktur, lebih singkat atau menyingkat koding dalam pengkodingan dan lebih sederhana.

Secara sederhana konsep MVC terdiri dari 3 bagian yaitu bagian pertama yaitu *Model*, lalu *View* dan yang terakhir adalah bagian *Controller*. Di dalam sebuah web yang dinamis paling tidak terdiri dari 3 hal utama yang menyusunya, yaitu basis data, logika aplikasi dan cara menampilkan halaman web. 3 hal itu direpresentasikan menggunakan MVC yaitu *model* untuk basis data lalu *controller* untuk logika aplikasi, dan *view* untuk menampilkan halaman *website* tersebut.

* + 1. **Xampp**

Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (localhost), yang terdiri atas program Apache, HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP danPerl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*) dan bebas (gratis) dan merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

* + 1. **MySQL**

MySQL adalah suatu software sistem manajemen database yang menggunakan standar SQL (*Structured Query Language*), yaitu bahasa standar yang paling banyak digunakan untuk mengakses database. Abdul Kadir (2009:15) mendefinisikan bahwa MySQL merupakan *software* yang tergolong *database server* dan bersifat *Open Source*.

*Open Source* menyatakan bahwa *software* ini di lengkapi oleh *source code* (kode yang di pakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi. Hal menarik lainnya adalah MySQL juga bersifat *multiplatform*. MySQL dapat dijalankan pada berbagai sistem operasi.

Keunggulan dari MySQL adalah :

1. MySQL dapat digunakan dan dimanfaatkan oleh banyak CPU sekaligus. MySQL mendukung penggunaan oleh beberapa *user* pada waktu yang bersamaan, oleh karena itu database *server* dapat diakses oleh *client* secara bersamaan pula.
2. MySQL merupakan *open source software*. Untuk menggunakan MySQL, yang mana merupakan lisensi dari GPL, *user* dapat mempergunakannya secara cuma-cuma, tanpa dipungut biaya.
3. MySQL mampu berjalan dalam berbagai sistem operasi (Portability). MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Microsoft Windows, Mac OS X server,Solaris, Amiga, Linux, FreBSD, HP-UX, dan lainnya.
4. Performance Tuning yang baik MySQL memiliki kecepatan yang sangat baik dalam menangani query sederhana.
5. Scalability and Column Types Support MySQL mampu menangani database yang besar dengan tipe kolom yang sangat kompleks.
6. High Security MySQL memiliki sistem sekuritas yang tinggi dengan disertai beberapa lapisan sekuriti seperti level subnetmask, nama host, dan ijin akses *user* dengan sistem perijinan yang disertai dengan serta password terenkripsi.
7. Standard Command and Function MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query, yang mana merupakan fungsi standar dari SQL.
8. Connectivity MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Named Pipes (NT), dan Unix soket (Unix).
9. Flexibility Table Structure MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan *database* lainnya seperti Oracle.
10. Localisation MySQL dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa.
11. Interface MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemrograman menggunakan fungsi API.
12. Clients dan Tools MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, yang mana pada setiap *tool* disertakan petunjuk *online*.
    * 1. **Web Service**

*Web service* adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum,*web service*  dapat diidentifikasikan dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service*.

Berbeda dengan URL *web* pada umumnya, URL *web service* hanya menggandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi.

*Web service* dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Menurut W3C *Web services Architechture Working Group* pengertian *Web Service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya WSDL).

Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar Web (Web Services Architecture Working Group, 2004). *Web* pada umumnya digunakan untuk melakukan *respon* dan *request* yang dilakukan antara *client* dan *server*. Sebagai contoh, seorang pengguna layanan *web* tertentu mengetikan alamat *url web* untuk membentuk sebuah *request*.

*Request* akan sampai pada *server*, diolah dan kemudian disajikan dalam bentuk sebuah respon. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana. Sedangkan pada *web service*hubungan antara *client*dan *server*tidak terjadi secara langsung. Hubungan antara*client*dan *server*dijembatani oleh file *web service*dalam format tertentu.

Sehingga akses terhadap *database e*akan ditanggani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service*. Peran dari *web service*ini akan mempermudah distribusi sekaligus integrasi database yang tersebar di beberapa *server* sekaligus.

* + 1. **Rest (*Representational State Transfer*)**

REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang sering diterapkan dalam pengembangan layanan berbasis web.

Arsitektur REST, yang umumnya dijalankan via HTTP (Hypertext Transfer Protocol), melibatkan proses pembacaan laman web tertentu yang memuat sebuah file XML atau JSON.

File inilah yang menguraikan dan memuat konten yang hendak disajikan. Setelah melalui sebuah proses definisi tertentu, konsumen akan bisa mengakses antarmuka aplikasi yang dimaksudkan. Kekhasan REST terletak pada interaksi antara klien dan server yang difasilitasi oleh sejumlah tipe operasional (verba) dan Universal Resource Identifiers (URIs) yang unik bagi tiap-tiap sumberdaya.

Masing-masing verba – GET, POST, PUT dan DELETE – memiliki makna operasional khusus untuk menghindari ambiguitas.

REST kerap dipergunakan dalam mobile application, situs web jejaring sosial, mashup tools, dan automated business processes. Arsitektur REST yang decoupled (terpisah) serta beban komunikasi yang ringan antara produsen dan konsumen membuatnya populer di dunia cloud-based API, seperti yang disajikan oleh Amazon, Microsoft, dan Google.

Layanan berbasis web yang menggunakan arsitektur REST semacam itu dinamakan RESTful APIs (Application Programming Interfaces) atau REST APIs.