



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:1) “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Sindu dan Paramartha (2019:2) mengatakan, “Komputer merupakan sebuah mesin penghitung elektronik yang cepat dapat menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan di memorinya (*stored program*) dan menghasilkan output informasi”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian komputer adalah alat elektronik yang dipakai untuk mengolah dan memproses data sehingga menghasilkan output berupa informasi.

2.1.2 Pengertian Perangkat Lunak

Sukanto dan Shalahuddin (2018:2) mengatakan, “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*)”.

Yurindra (2017:1), “Perangkat lunak adalah transformasi informasi yang memproduksi, mengatur, memperoleh, memodifikasi, menampilkan, atau memancarkan informasi sehingga pekerjaan dapat menjadi lebih sederhana”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian perangkat lunak adalah program komputer yang dapat melaksanakan tugas sesuai kehendak pemakai.



2.1.3 Pengertian Basis Data (*Database*)

“Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.” (Sukamto dan Shalahuddin, 2018:43).

Menurut Lubis (2016:2), “Basis Data merupakan gabungan *file* data yang dibentuk dengan hubungan/relasi yang *login* dan dapat diungkapkan dengan catatan serta bersifat independen”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian basis data suatu media untuk menyimpan data agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2.1.4 Metode Pengembangan Aplikasi

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2018:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada



implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

2.2 Teori Khusus


2.2.1 *Unified Modelling Language (UML)*

Menurut Sukamto dan Shalahuddin (2018:133), “UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

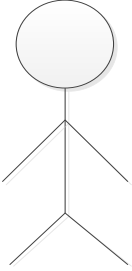


2.2.2 *Use Case Diagram*

Sukamto dan Shalahuddin (2018:155) mengatakan, “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem”. Adapun simbol-simbol yang akan digunakan dalam *use case* adalah sebagai berikut:

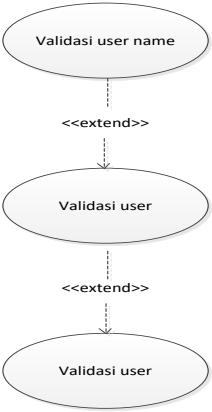

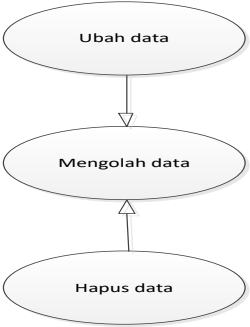
Tabel 2.1 Simbol-simbol *Use case Diagram*

No.	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .

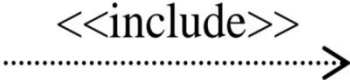
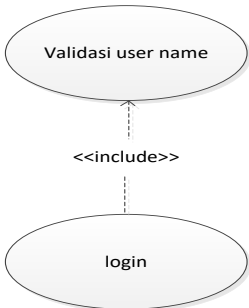
Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

No.	Simbol	Deskripsi
2.	<p data-bbox="395 421 564 450">Aktor / <i>actor</i></p>  <p data-bbox="560 824 707 853">nama aktor</p>	<p data-bbox="903 421 1390 891">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
3.	<p data-bbox="395 918 687 947">Assosiasi / <i>association</i></p> 	<p data-bbox="903 918 1390 1115">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4.	<p data-bbox="395 1205 603 1234">Extensi / <i>extend</i></p> 	<p data-bbox="903 1205 1390 1787">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal :</p>

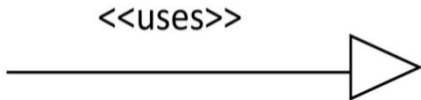
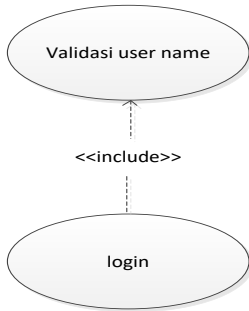
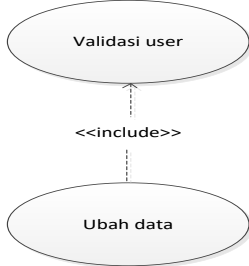
Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

No.	Simbol	Deskripsi
		 <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya</p>

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

No.	Simbol	Deskripsi
		(umum).
6.	Menggunakan / <i>include</i> / <i>uses</i> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :  <ul style="list-style-type: none"> - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut:

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*



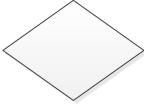

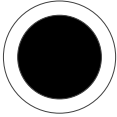
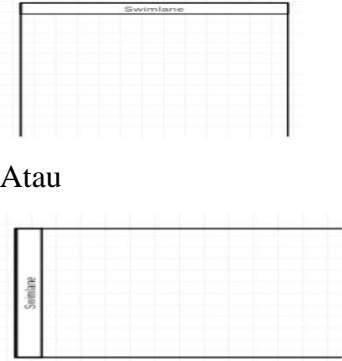
No.	Simbol	Deskripsi
		 <p data-bbox="919 766 1393 1070">- <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut:</p>  <p data-bbox="919 1402 1393 1594">Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto dan Shalahuddin (2018:155)

2.2.3 Activity Diagram

Sukamto dan Shalahuddin (2018:161) menyatakan, “*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane  Atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

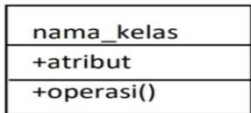






Sumber : Sukamto dan Shalahuddin (2018:155)



2.2.4 Class Diagram

Sukamto dan Shalahuddin (2018:141) menyatakan, “*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) .
6.	Keberuntungan/ <i>depedensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agrgasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

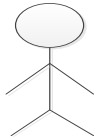
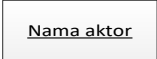

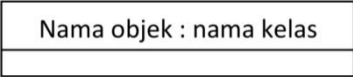

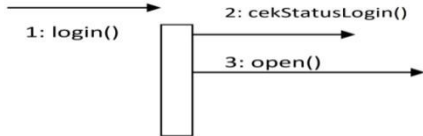
Sumber : Sukamto dan Shalahuddin (2018:141)




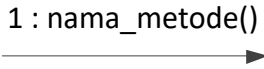
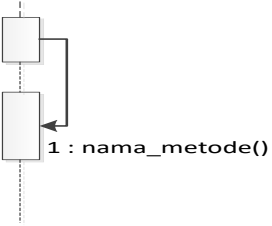
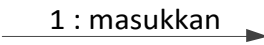

2.2.5 Sequence Diagram

Sukamto dan Shalahuddin (2018:165) menyatakan, “*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu daur hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

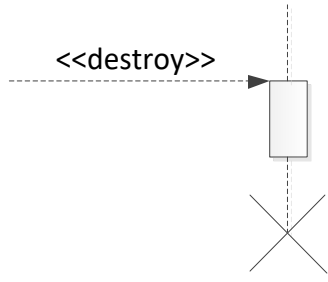
Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama actor</p> <p>Atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
		Maka cek StatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau sendirinya,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
		yang menerima kembalian.
9.		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Sukamto dan Shalahuddin (2018:165)

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Indrajani (2018:3), “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

Marjito dan Tesaria (2016:42) “Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju”.

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa aplikasi adalah suatu program komputer yang dibuat untuk menyelesaikan tugas-tugas untuk mencapai sasaran yang akan dituju.

2.3.2 Pengertian Pengelolaan Data

Sutabri (2012:169), “Pengelolaan Data adalah suatu proses menerima data sebagai masukan (input), memproses (processing) menggunakan program tertentu, dan mengeluarkan hasil proses data tersebut dalam bentuk informasi (output)”.



Kristanto (2018:8), “Pengelolaan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa pengelolaan data adalah sebuah proses yang mengolah data mentah menjadi sebuah informasi yang berguna.

2.3.3 Pengertian TK/TP Al-Qur’an

Taman Kanak-kanak Al-Quran (TKA) adalah lembaga/sekolah yang mendidik anak usia 4-6 tahun sehingga dapat memahami dan mengamalkan Al-Quran. Sedangkan Taman Pendidikan Al-Quran (TPA) adalah lembaga/sekolah yang mendidik anak usia 7-12 tahun sehingga mampu memahami dan mengamalkan Al-Quran (Agusalim, 2015).

2.3.4 Pengertian Web

Abdulloh (2018:1), “*Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia”.

Rerung (2018:1) mendefinisikan bahwa “*web* adalah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *hypertext transfer protocol*”.

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa *website* adalah aplikasi internet yang berisi informasi data digital yang disediakan melalui jalur internet sehingga dapat diakses dan dilihat semua orang di dunia.

2.3.5 Pengertian Aplikasi Pengelolaan Data TK/TP Al-Qur’an berbasis Web pada BKPRMI Prabumulih

Aplikasi pengelolaan data TK/TP Al-Qur’an adalah sebuah program komputer yang dibuat untuk mengolah data mentah menjadi sebuah informasi yang bermanfaat, terutama dalam melakukan perekapan data TK/TP Al-Qur’an.



2.4 Teori Program

2.4.1 Pengertian XAMPP

Elgamar (2020:11) menyatakan bahwa, XAMPP merupakan sejenis paket aplikasi yang didalamnya terdapat beberapa perangkat lunak lain yang mendukung dalam pengolahan website atau aplikasi berbasis web.

Aryanto (2016:4), “Xampp merupakan sebuah aplikasi perangkat lunak pemrograman dan database yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti : Apache, HTTP, MySQL, database, bahasa pemrograman PHP dan Perl”.

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa XAMPP adalah aplikasi pemrograman dan database yang terdiri dari Apache, HTTP, MySQL, database, bahasa pemrograman PHP dan Perl agar dapat mengakses web dinamis secara lokal.



Gambar 2.1 Tampilan XAMPP

2.4.2 PHP

2.4.2.1 Pengertian PHP

Supono dan Putratama (2018:3), “PHP (PHP : *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadikode mesin yang dapat dimengerti oleh computer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML.”

Madcoms(2016:2), “PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML”.



Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa PHP adalah bahasa pemrograman yang disisipkan ke dalam HTML agar dapat berjalan dalam web server.



Gambar 2.2 Tampilan Logo PHP

2.4.2.2 Tipe Data PHP

Tipe data dapat dikelompokkan menjadi beberapa bagian. Supono dan Putratama (2018:22), menjelaskan beberapa tipe data dalam PHP, yaitu sebagai berikut.

1. Tipe Data Integer (Bilangan Bulat)

Tipe Data *Integer* adalah tipe data yang mempresentasikan bilangan bulat seperti 3,6,3000,4596 dan sebagainya. Rentang Nilai yang diperbolehkan pada tipe data *integer* adalah dari 2,147,483,648 sampai dengan +2,147,483,647. Nilai dari tipe data *integer* dapat ditulis dalam bentuk decimal, octal, maupun heksadesimal.

2. Tipe *Floating – Point* (Bilangan Rill)

Tipe *Floating-point* adalah tipe data yang mempresentasikan nilai *numeric* dalam bentuk pecahan atau mengandung angka decimal belakangan koma. Nilai yang diperbolehkan pada tipe *floating-point* adalah 1.7E-308 sampai dengan 1.7E+308 dengan 15 digit akurasi.

3. Tipe Data Karakter

Tipe data karakter adalah untuk merepresentasikan data dengan nilai karakter tunggal, contohnya: 'A', 'I', 'U', 'E', 'O', dan sebagainya.

4. Tipe Data String

Tipe data *string* mempresentasikan data yang berupa teks (kumpulan karakter).



5. Tipe Data Array

Tipe array merupakan suatu variable yang menyimpan sekelompok nilai yang memiliki sifat yang sama, yang dapat diidentifikasi atau diakses berdasarkan posisi atau dengan nama yang telah didefinisikan sebelumnya.

6. Tipe Data Object

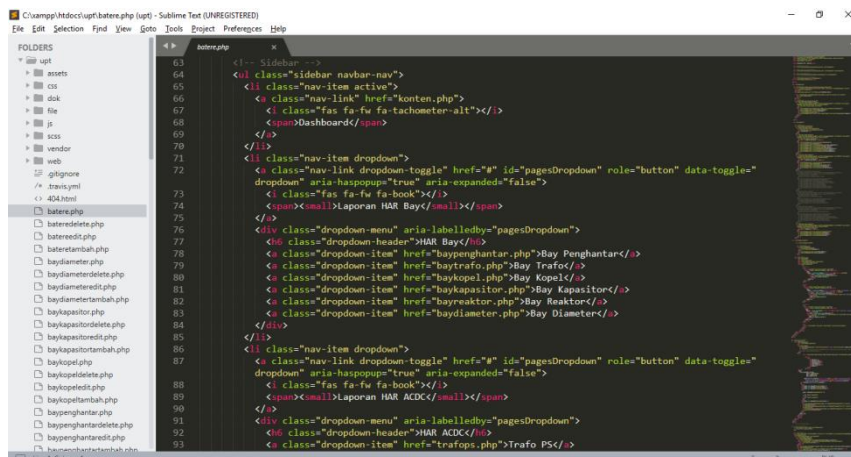
Tipe data object dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data object bisa berupa bilangan, variable atau fungsi.

2.4.3 Pengertian Sublime Text

Rerung (2018:25), “Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform operating sistem dengan menggunakan teknologi API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah fleksibel dan powerfull.”

“Sublime Text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi” (Supono dan Putratama, 2018:14).

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa sublime text adalah aplikasi editor yang dapat berjalan di beberapa platform untuk membuat atau meng-edit suatu aplikasi.



Gambar 2.3 Tampilan Sublime Text



2.4.4 Pengertian MySQL

Sukamto dan Shalahuddin (2018:46), “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

Hidayatullah dan Kawistara (2017:175) “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi web”.

Berdasarkan kedua definisi diatas, maka penulis dapat menyimpulkan bahwa MySQL adalah pemrograman yang digunakan untuk mengelola data pada RDBMS.