

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Pendukung Keputusan

Definisi awal Sistem Pendukung Keputusan sebagai sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi struktur [5]. SPK dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka.

SPK ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma. Definisi awal mengimplikasikan (tidak menyatakan secara spesifik) bahwa sistem akan berbasis komputer, akan beroperasi online interaktif, dan kemungkinan akan memiliki kapabilitas *input* grafis. Definisi awal terbuka terhadap beberapa interpretasi. Sistem pendukung keputusan merupakan bagian dari sistem informasi berbasis komputer yang mengatasi masalah ini.

Termasuk sistem berbasis pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi, perusahaan, atau lembaga pendidikan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah kompleks. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif digunakan oleh pemakai [6]. Dengan dibangunnya sebuah sistem pendukung keputusan yang terkomputerisasi, maka subjektifitas dalam pengambilan keputusan

dapat dikurangi dan diganti dengan pelaksanaan seluruh kriteria. Sehingga peserta yang terbaik yang akan terpilih.

2.1.1 Karakteristik Sistem Pendukung Keputusan

Dibawah ini yang merupakan karakteristik dari sistem pendukung keputusan yaitu.

1. Sistem pendukung keputusan dalam pemberian dukungan untuk pengambilan keputusan dalam keadaan semi terstruktur dan tak berstruktur yang berasal dari pertimbangan manusia dan informasi komputerisasi.
2. Suatu dukungan individu dan kelompok.
3. *User* mudah dalam penggunaannya.
4. Dukungan pada semua level manajerial dari eksekutif puncak hingga manajerial ini.
5. Dukungan dalam semua *fase* pada proses pengambilan keputusan yaitu berupa *intelligence, design, choice* dan *implementation*.
6. Adanya peningkatan efektivitas dari pengambilan keputusan dari pada efisiensi.
7. Suatu adaptivitas sepanjang waktu.
8. Suatu akses yang berfungsi untuk berbagai sumber daya, tipe dan format mulai sistem informasi hingga sistem berorientasi objek.
9. Suatu kendali yang penuh untuk pengambilan keputusan.

2.1.2 Tujuan Sistem Pendukung Keputusan

Sistem pendukung keputusan memiliki tujuan yaitu sebagai berikut :

1. Meningkatkan tingkat efektif keputusan yang diambil manejer sehingga lebih efisien
2. Mampu memberikan suatu dukungan dan pertimbangan manajer tanpa mengabaikan fungsi manajer.

3. Mampu membantu manajer untuk pengambilan keputusan dalam upaya menangani masalah yang semi terstruktur.
4. Mampu berdaya saing.
5. Memiliki dukungan yang berkualitas.
6. Mampu mengatasi keterbatasan yang kognitif pada suatu pemrosesan dan penyimpanan.

2.1.3 Tahapan Dalam Pengambilan Keputusan

Adapun tahapan-tahapan dalam mengambil keputusan dalam sistem pendukung keputusan yaitu :

1. *Intelligence Phase*

Dalam tahap ini adanya suatu proses penelurusan pada permasalahan serta adanya proses pengenalan masalah lalu mengidentifikasi suatu masalah.

2. *Design Phase*

Dalam tahap ini adanya proses pengembangan dan solusi yang menjadi bahan pertimbangan sehingga dibutuhkan proses memvalidasi atau pun memverifikasi dalam upaya peningkatan keakuratan.

3. *Choise Phase*

Dalam tahap ini adanya pemilihan berbagai solusi untuk menentukan suatu pilihan yang melibatkan kriteria dan tujuan.

4. *Implementation Phase*

Dalam tahap ini merupakan rancangan sistem dan sangat berpengaruh pada aktifitas pengambilan keputusan dalam penyelesaian masalah dengan tepat.

2.1.4 Manfaat Sistem Pendukung Keputusan

Pada sistem pendukung keputusan memiliki manfaat sebagai berikut :

1. Sistem pendukung keputusan mampu mengambil suatu keputusan dalam pemrosesan data ataupun informasi lainnya.
2. Sistem pendukung keputusan mampu memecahkan masalah.
3. Sistem pendukung keputusan mampu menghasilkan solusi lebih cepat.
4. Sistem pendukung keputusan juga mampu menjadi stimulan terhadap pengambilan keputusan untuk pemahaman persoalan yang ada.

Dengan adanya sistem pendukung keputusan dalam penentuan pertukaran Pelajar maka proses penyeleksian menjadi lebih efektif dan meminimalisir terjadinya kesalahan atau kurang tepatnya pengambilan keputusan [5].

2.2 AHP

AHP dapat digunakan untuk merangsang timbulnya gagasan untuk melaksanakan tindakan kreatif, dan untuk mengevaluasi keefektifan tindakan tersebut. Selain itu, untuk membantu para pemimpin menetapkan informasi apa yang patut dikumpulkan guna mengevaluasi pengaruh faktor-faktor relevan dalam situasi kompleks. AHP juga dapat melacak ketidak konsistenan dalam pertimbangan dan preferensi peserta, sehingga para pemimpin mampu menilai mutu pengetahuan para pembantu mereka dan pemantapan pemecahan itu.

Adapun algoritma penyelesaian metode *Analytical Hierarchy Process* (AHP) yaitu sebagai berikut [7].

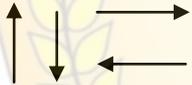
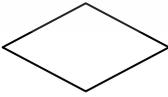
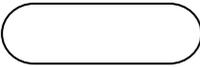
1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolak ukur penyelesaian masalah dan menentukan tingkat kepentingan dari setiap kriteria.
2. Langkah 2 : Menghitung Nilai Matriks Perbandingan dari masing-masing kriteria berdasarkan tabel nilai kepentingan

3. Langkah 3 : Menghitung nilai bobot kriteria(W_j)
4. Langkah 4 : Menghitung nilai Consistency Indeks
5. Langkah 5 : Menghitung nilai Consistency Ratio

2.3 Flowchart

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. Adapun simbol-simbol yang digunakan secara dasar adalah sebagai berikut [8]:

Tabel 2.1 Simbol - Simbol *Flowchart*

No	Simbol	Nama	Deskripsi
1		<i>Input-Output</i>	Digunakan untuk mewakili data <i>input/output</i> .
2		Proses	Digunakan untuk mewakili suatu proses.
3		Alir/Arah	Digunakan untuk menunjukkan Arah/Alir dari suatu proses.
4		Penghubung	Digunakan untuk menunjukkan sambungan dari aliran yang terputus di halaman yang sama.
5		Keputusan	Digunakan untuk suatu penyelesaian kondisi dalam program.
6		Awal/Akhir (Terminator),	Digunakan untuk menunjukkan awal dan akhir dari proses.
7		Persiapan	Digunakan untuk memberikan nilai awal dari proses.
8		<i>Predefined</i> Proses	Digunakan untuk proses yang detilnya terpisah.

2.4 Pemodelan Sistem

Pemodelan adalah gambaran dengan aturan tertentu dari kenyataan yang sederhana dan dibuat dalam bentuk pemetaan. Pemodelan perangkat lunak yang baik digunakan pada pengembangan sistem informasi sehingga dapat terencana. Adapun jenis pemodelan sistem adalah sebagai berikut :

2.4.1 UML (Unified Modelling Language)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement* (tata syarat), membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” [9]. Manual Referensi UML disusun menjadi tiga bagian: ikhtisar sejarah UML dan pemodelan, survei konsep UML, dan ensilopedia alfabet istilah dan konsep UML [10]. *Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek [11].

Pada teknik pemrograman berorientasi objek, muncul lah sebuah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang di bangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Dalam UML secara umum digambarkan dalam bentuk *use case diagram*, *activity diagram* dan *class diagram*.

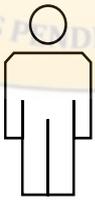
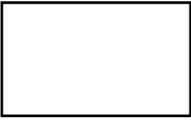
Tujuan Penggunaan *Unified Modelling Language* adalah sebagai berikut [12].

1. Analisis dan desain pandangan statis aplikasi
2. Menjelaskan tanggung jawab suatu sistem
3. Basis untuk komponen diagram dan penyebaran
4. *Forward and reverse engineering.*

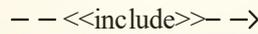
2.4.1.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [9].

Tabel 2.2 Simbol - Simbol *Use Case Diagram*

No	Nama	Simbol	Keterangan
1	<i>Use case</i>		Fungsional yang disediakan dari sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> .
2	Aktor / <i>actor</i>		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang.
3	<i>System Boundary</i>		Digambarkan dengan kotak di sekitar <i>use case</i> . Biasanya digunakan apabila memberikan beberapa <i>alternative system</i> yang dapat dijadikan pilihan. <i>System boundary</i> dalam penggunaannya optimal.

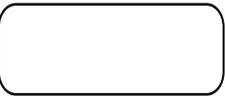
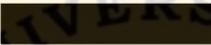
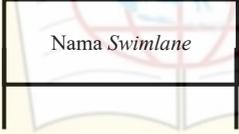
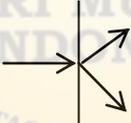
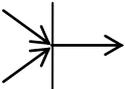
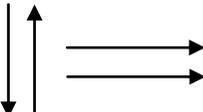
Tabel 2.2 Simbol - Simbol *Use Case Diagram* (Lanjutan)

No	Nama	Simbol	Keterangan
4	Asosiasi / <i>association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
5	Extensi / <i>extend</i>		Relasi <i>use case</i> tambahan ke semua <i>use case</i> yang ditambahkan dapat berdiri sendiri, mirip dengan prinsip <i>inheritance</i> .
6	Generalisasi / <i>generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
7	Menggunakan / <i>include / uses</i>		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

2.4.1.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem [9].

Tabel 2.3 Simbol - Simbol *Activity Diagram*

No	Nama	Simbol	Keterangan
1	Status Awal		Status awal aktifitas sistem, sebuah diagram memiliki sebuah status awal.
2	Aktifitas		Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>Decision</i>		Asosiasi percabangan jika ada pilihan aktifitas lebih dari satu.
4	Penggabungan / <i>Join</i>		Asosiasi penggabungan lebih dari satu aktifitas digabungkan satu.
5	Status akhir		Status akhir yang dilakukan sistem sebuah diagram aktifitas memiliki sebuah status akhir.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
7	Percabangan / <i>Fork</i>		Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau menggabungkan dua kegiatan paralel menjadi satu
8	<i>Join</i> / Penggabungan		Digunakan untuk menggabungkan <i>parallel</i> yang lebih dari satu menjadi satu.
9	<i>Line Connection</i>		Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

Sebuah *activity diagram* memiliki elemen-elemen diantaranya sebagai berikut [13].

1. Sebuah keadaan awal (*start state*) dan keadaan akhir (*end state*).

2. Aktivitas-aktivitas yang menggambarkan satu tahapan dalam *workflow* tersebut. Transisi yang menggambarkan keadaan apa yang mengikuti suatu keadaan lainnya.
3. Keputusan (*decision*), elemen yang menyediakan pilihan alur dalam *workflow*.
4. Batang penyelarasan (*synchronization bar*) memperlihatkan sub alur paralel.
5. *Swimlane* yang menjelaskan pemeran bisnis yang bertanggung jawab terhadap aktivitas yang dikandungnya

2.4.1.3 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian. Kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan *variable-variable* yang dimiliki oleh suatu kelas-kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek. *Class* memiliki tiga area pokok yaitu [14].

1. Nama, kelas harus mempunyai sebuah nama.
2. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki
3. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

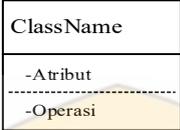
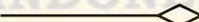
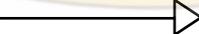
Tujuan dari *class diagram* dalam perancangan *table database* sebagai berikut [15].

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

3. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

Adapun simbol-simbol *class diagram* sebagai berikut :

Tabel 2.4 Simbol - Simbol *Class Diagram*

No	Nama	Simbol	Keterangan
1	Kelas		Kelas pada struktur <i>system</i>
2	Antarmuka / <i>Interface</i>		Sama dengan konsep <i>interfaced</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 lain.
5	Agregasi / <i>Aggregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)
6	Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7	Kebergantungan / <i>Dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas

Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*. Adapun simbol sebagai berikut [16].

Tabel 2.5 *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

Visibility merupakan *property* yang sangat penting dalam pendefinisian atribut dan operasi pada suatu *class*. Adapun simbol *visibility* sebagai berikut, yaitu [16]:

Tabel 2.6 *Visibility Class Diagram*

Visibility	Penjelasan
<i>public</i> (+)	Dapat diakses oleh <i>class</i> lain. Dilambangkan dengan tanda +.
<i>protected</i> (#)	Hanya dapat diakses oleh <i>class</i> itu sendiri dan <i>class</i> turunannya (<i>sub class</i>)
<i>private</i> (-)	Hanya dapat diakses oleh <i>class</i> itu sendiri

2.5 Perangkat Pendukung

Pada penelitian ini akan mengembangkan sebuah aplikasi dalam mengimplementasikan nya dalam kehidupan sehari-sehari. sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*) [17]. Adapun aplikasi perangkat pendukung dalam menjalankan suatu sistem antara lain:

2.5.1 Microsoft Visio

Microsoft Visio (atau sering disebut *Visio*) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh *Microsoft*

Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagram nya [18].

Terlepas dari apa yang mungkin terpikirkan dari hubungannya dengan aplikasi *Microsoft Office* lainnya, dengan *Visio* rasanya seperti di rumah, terutama karena tujuan dari keterkaitan *user interface* gaya presentasi visual dari kelompok yang terkait fungsi, dan *Visio* termasuk di dalamnya, pertama dan terutama, sebuah produk visual. Untuk menggambarkan model banyak alat bantu yang bisa digunakan. Bahkan pengolah kata saat ini, misalnya *Microsoft Word*, menyediakan juga *tool* untuk menggambar pada menu “*design*”. Salah satu aplikasi yang cukup baik untuk membuat bagan adalah *Microsoft Visio* yang dijual terpisah dengan *Microsoft Office* (*word, excel, access*, dan pendukung lainnya). Postingan kali ini sekedar *sharing* bagaimana membuat bagan sederhana dengan *Visio*.

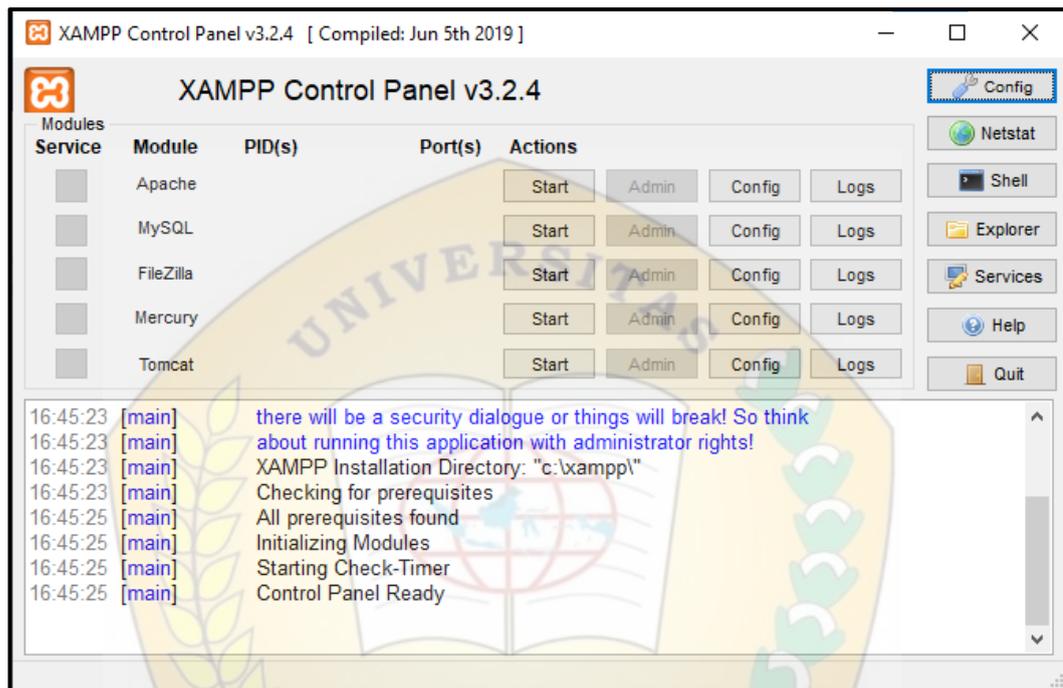


Gambar 2.1 *Microsoft Visio*

2.5.2 XAMPP

Web server adalah suatu program komputer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari komputer klien, yang dikenal dengan nama *web browser* dan melayani mereka dengan menyediakan *repon*

HTTP berupa konten data [19]. *Web* bisa diartikan juga Sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, *video*, *audio*, dan animasi lainnya yang disediakan melalui jalur koneksi *internet* [20]. Adapun Gambar tampilan awal XAMPP sebagai berikut.



Gambar 2.2 XAMPP

Ada beberapa jenis *software* untuk membangun *web server local* atau *localhost* yang *support* sistem operasi *windows* di antaranya adalah Xampp dan Mysql. XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi. Langkah-langkah pembuatan dimulai dari instalasi paket XAMPP, kemudian perancangan database MySQL, dan pembuatan *file-file* eksekusi yang disimpan dalam *file* ekstensi PHP. XAMPP mengkombinasi beberapa paket perangkat lunak berbeda ke dalam satu paket. Paket yang disediakan antara lain adalah Apache, MySQL, PHP, FileZilla FTP *Server*, phpMyAdmin, dan lain-lain [21].

2.5.3 Sublime Text

Sublime Text merupakan salah satu *text* editor yang sangat *powerful* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi[19]. *Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*[22]. Adapun Gambar tampilan awal *Sublime Text* sebagai berikut.



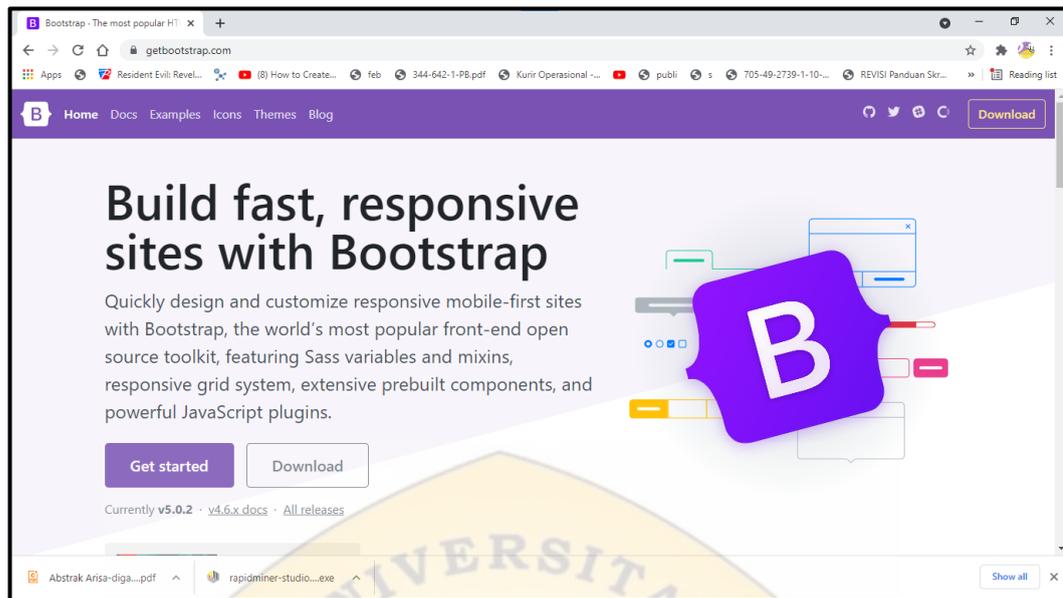
Gambar 2.3 *Sublime Text*

2.5.4 Bootstrap

Bootstrap adalah *front-end framework* yang bagus dan luar biasa yang mendapatkan tampilan untuk *mobile device* (*Handphone, smartphone* dll.) guna mempercepat dan mempermudah pengembangan *website*.

Bootstrap menyediakan HTML, CSS dan *Javascript* siap pakai dan mudah untuk dikembangkan. Dengan *Bootstrap* kita juga bisa membangun *web* dinamis

atau pun statis [23]. Adapun Gambar tampilan awal *Bootstrap* sebagai berikut.



Gambar 2.4 *Bootstrap*

