

BAB II

LANDASAN TEORI

2.1 KONSEP DASAR SISTEM

2.1.1 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan. Japerson .H (2015:13)

2.1.2 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. Tata Sutabri (2012:47)

1. Blok masukan (*input block*)

Input mewakili data yang masuk kedalam sistem informasi. Input yang dimaksud adalah metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan “tool box” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknis (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

6. Blok kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, keagalan-kegagalan sistem itu

sendiri, ketidak efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.1.3 Fungsi Sistem Informasi

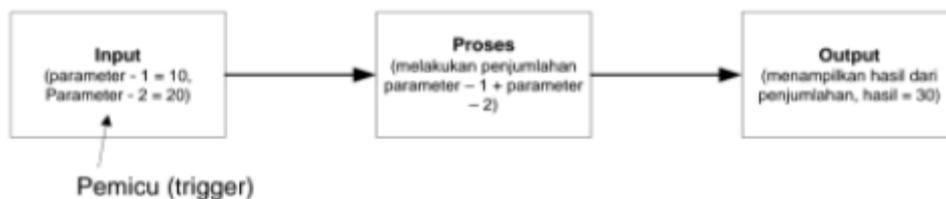
Adapun fungsi dari sistem informasi yang sangat bermanfaat di era teknologi seperti ini menurut E. Y. Anggraeni (2017:2) yaitu :

1. Untuk meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
2. Memperbaiki produktivitas aplikasi pengembangan dan pemeliharaan sistem.
3. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan pada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

2.1.4 Proses Sistem Informasi

Sistem memerlukan inputan dari pengguna sistem. Inputan tersebut akan dijadikan *parameter* sebagai bahan baku untuk pengolahan data. Proses penginputan *parameter* oleh pengguna sistem biasanya disebut proses *triggering* (pemicu sistem). Tanpa pemicu sistem tidak akan berjalan. Pemicu sistem bisa berupa orang (manusia), mesin ataupun sistem lain yang terintegrasi. Sebagai contoh kita ambil dari ilustrasi sistem komputer, dimana *processor* tidak akan bekerja jika data tidak diinputkan oleh *user*. Prof. Dr. Sri Mulyani, Ak., CA (2017:10)

Setelah sistem menerima inputan dari user, maka sistem akan memproses data tersebut sesuai dengan perintah ataupun program yang sudah ditanamkan dalam sistem. Kemudian sistem akan memberikan output dari hasil pengolahan data yang sudah diinputkan user tersebut. Prof. Dr. Sri Mulyani, Ak., CA (2017:10).



Gambar 2.1 Proses sistem informasi

2.1.5 Pengertian Perancangan

Perancangan atau merancang merupakan suatu usaha untuk menyusun, mendapatkan, dan menciptakan hal-hal baru yang bermanfaat bagi kehidupan manusia. Dalam hal ini, merancang dapat yang benar-benar baru atau pengembangan produk yang sudah ada, sehingga mendapat peningkatan kinerja dari produk tersebut. Purna Irawan (2017:3)

2.2 JUMANTIK

2.2.1 Pengertian Jumentik

Kader juru pemantau jentik (jumentik) adalah kelompok kerja kegiatan pemberantasan penyakit demam berdarah dengue di tingkat Desa dalam wadah Lembaga Ketahanan Masyarakat Desa (Depkes RI, 1992). Tujuan Menggerakkan peran serta masyarakat dalam usaha pemberantasan penyakit DBD, terutama dalam pemberantasan jentik nyamuk penularnya sehingga penularan penyakit demam berdarah dengue di tingkat desa, dapat dicegah atau dibatasi.

Menurut Depkes RI (2005) peran kader kesehatan dalam menanggulangi DBD adalah:

- a. Sebagai anggota PJB di rumah-rumah dan tempat umum.
- b. Memberikan penyuluhan kepada keluarga dan masyarakat.
- c. Mencatat dan melaporkan hasil PJB Kepala Dusun atau Puskesmas secara rutin minimal setiap minggu dan bulanan.
- d. Mencatat dan melaporkan kasus kejadian DBD kepada RW/Kepala Dusun atau Puskesmas.
- e. Melakukan PSN dan pemberantasan DBD secara sederhana seperti pemberian bubuk abate dan ikan pemakan jentik.

2.2.2 Survei Penduduk

Survei Penduduk adalah cara pengumpulan data yang dilaksanakan melalui pencacahan sampel dari suatu populasi untuk memperkirakan karakteristik objek pada saat tertentu. Dilakukan secara sampling terhadap sekelompok penduduk yang

dianggap wakil dari seluruh populasi, untuk memperoleh keterangan yang bersifat khusus dan lebih terperinci. Tingkat representatif data survei relatif rendah dibandingkan data sensus, namun keterangan yang diperoleh lebih terperinci. Suwito (2020:9).

2.3 METODE DEKRIPTIF (SATU SAMPEL)

Pengujian hipotesis deskriptif pada dasarnya merupakan proses pengujian generalisasi hasil penelitian yang didasarkan pada satu sampel. Kesimpulan yang dihasilkan nanti adalah apakah hipotesis yang diuji itu dapat digeneralisasikan atau tidak. Bila H_0 diterima berarti dapat digeneralisasikan. Dalam pengujian ini variabel penelitiannya bersifat mandiri, oleh karena itu hipotesis penelitian tidak berbentuk perbandingan ataupun hubungan antar dua variabel atau lebih.

Secara skematis pengujian hipotesis deskriptif terdapat beberapa macam teknik statistik yang dapat digunakan untuk menguji hipotesis tersebut. Teknik statistik mana yang akan dipakai tergantung pada jenis data yang akan dianalisis.

Tabel 2.1 Statistik Yang Digunakan Untuk Menguji Hipotesis Deskriptif.

Jenis/ Tingkatan Data	Teknik Statistik Yang Digunakan Untuk Pengujian
Nominal	1. Test Binominal 2. Chi Kuadrat (1 Sampel)
Ordinal	3. Run test
Menurut Interval/ Ratio	4. T-test (1 sampel)

2.4 EXTREME PROGRAMING (XP)

Kent Beck merupakan orang yang menciptakan metode XP selama ia bekerja di proyek Chrysler Comprehensive Compensation (C3). Extreme Programming (XP) adalah sebuah pendekatan atau metode pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. Tahapan-tahapan yang harus dilalui dengan XP yaitu, tahapan eksplorasi, perencanaan, iterasi pengembangan sistem, dan tahap produksi akhir. Laboratorium Rekayasa Perangkat Lunak (2013).

Adapun 12 aspek pada XP pada tahapan-tahapan yang harus dikerjakan, diantaranya (Widhiartha, 2008:4):

1. The Planning Game Pada aspek ini mirip dengan metode RAD (Rapid Application Development) dengan proses yang singkat dan cepat, dengan tetap mengutamakan aspek teknik memisahkan unsur bisnis dengan unsur teknis dan pertemuan intensif antara klien dan pengembang perangkat lunak (software developing).
2. Small Releases, lingkup kegiatan release yang dilakukan dengan sekecil mungkin pada XP. Setiap developer menyelesaikan sebuah unit atau bagian dari perangkat lunak maka hasil tersebut harus segera dipresentasikan dan didiskusikan dengan klien. Jika memungkinkan untuk menerapkan unit tersebut pada perusahaan, hal itu juga dapat dilakukan sekaligus sebagai tes awal dari penerapan keseluruhan sistem.
3. Metaphor pada dasarnya sama dengan arsitektur perangkat lunak. Keduanya menggambarkan visi yang luas terhadap tujuan dari pengembangan perangkat lunak. Arsitektur yang saat ini banyak berisi diagram dan kode semacam UML

dianggap terlalu rumit untuk dimengerti, terutama oleh klien. Metaphor, walaupun mirip dengan arsitektur lebih bersifat naratif dan deskriptif.

4. Simple Design Pada XP desain dibuat dalam lingkup kecil dan sederhana. Tidak perlu melakukan antisipasi terhadap berbagai perubahan di kemudian hari. Dengan desain yang simpel apabila terjadi perubahan maka membuat desain baru untuk mengatasi perubahan tersebut dapat dengan mudah dilakukan dan resiko kegagalan desain dapat diperkecil.

5. Refactoring adalah salah satu aspek paling khas dari XP. Refactoring sendiri sangat sesuai untuk menjadi bagian XP karena Refactoring mengusung konsep penyederhanaan dari proses desain maupun struktur baris kode program. Dengan Refactoring tim pengembang dapat melakukan berbagai usaha untuk meningkatkan kualitas program tanpa kembali mengulang-ulang proses desain..

6. Testing XP menganut paradigma berbeda dalam hal tes dengan model pengembangan perangkat lunak lainnya. Jika pada pengembangan perangkat lunak lainnya tes baru dikembangkan setelah perangkat lunak selesai menjalani proses coding maka pada XP tim pengembang harus membuat terlebih dahulu tes yang hendak dijalani oleh perangkat lunak. Berbagai model tes yang mengantisipasi penerapan perangkat lunak pada sistem dikembangkan terlebih dahulu. Saat proses coding selesai dilakukan maka perangkat lunak diuji dengan model tes yang telah dibuat tersebut. Pengetesan akan jauh lebih baik apabila dilakukan pada setiap unit perangkat lunak dalam lingkup sekecil mungkin daripada menunggu sampai seluruh perangkat lunak selesai dibuat. Dengan memahami tahap ini kita dapat melihat bahwa siklus pada XP adalah requirement analysis -> test -> code -> design.

7. Pair Programming adalah melakukan proses menulis program dengan berpasangan. Dua orang programmer saling bekerjasama di komputer yang sama untuk menyelesaikan sebuah unit. Dengan melakukan ini maka keduanya selalu dapat berdiskusi dan saling melakukan koreksi apabila ada kesalahan dalam penulisan program. Aspek ini mungkin akan sulit dijalankan oleh para programmer yang memiliki ego tinggi dan sering tidak nyaman untuk berbagi komputer bersama rekannya.

8. Collective Ownership XP menuntut para programmer untuk berbagi pengetahuan untuk tiap baris program bahkan beserta hak untuk mengubahnya. Dengan pemahaman yang sama terhadap keseluruhan program, ketergantungan pada programmer tertentu ataupun berbagai hambatan akibat perbedaan gaya menulis program dapat diperkecil. Pada level yang lebih tinggi bahkan dimungkinkan para programmer dapat bertukar unit yang dibangunnya.

9. Coding Standards Dengan adanya coding standards yang telah disepakati terlebih dahulu maka pemahaman terhadap program akan menjadi mudah untuk semua programmer dalam tim. Hal ini dapat diterapkan sebagai contoh pada penamaan variabel dan penggunaan tipe data yang sama untuk tiap elemen semua record atau array pada program.

10. Continuous Integration Melakukan build setiap hari kerja menjadi sebuah model yang disukai oleh berbagai tim pengembang perangkat lunak. Hal ini terutama didorong oleh keberhasilan penerapan sistem ini oleh Microsoft dan telah sering dipublikasikan. Dengan melakukan build sesering mungkin berbagai kesalahan pada program dapat dideteksi dan diperbaiki secepat mungkin. Apabila banyak tim

pengembang perangkat lunak meyakini bahwa build sekali sehari adalah minimum maka pada XP hal tersebut adalah maksimum. Pada XP tim disarankan untuk melakukan build sesering mungkin misalnya setiap 4 jam atau bahkan lebih cepat lagi.

11. 40-hours Week Bekerja 8 jam sehari dan 5 hari seminggu adalah maksimal untuk tiap programmer, jika lebih dari itu programmer akan mengalami berbagai error program karena kelelahan. Hal tersebut dikemukakan oleh Beck orang yang menciptakan metode XP.

12. On-Site Customer Sebuah pendekatan klasik, di mana XP menganjurkan bahwa ada anggota dari klien yang terlibat pada proses pengembangan perangkat lunak. Yang lebih penting lagi ia harus ada di tempat pemrograman dan turut serta dalam proses build dan test yang dilakukan. Apabila ada kesalahan dalam pengembangan diharapkan klien dapat segera memberikan masukan untuk koreksinya.

2.5 PERALATAN PENDUKUNG (TOOLS SYSTEM)

2.5.1 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) Menurut Rosa dan Shalahuddin [2014, Hal.133], berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requerement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Menurut A. C. Prof dan Dr. Sri Mulyani NS (2016, Hal.48) mengatakan UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”. Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek. UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya menurut Rosa dan Shalahuddin (2014, Hal.155):

1. ***Use Case Diagram***, merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.
2. ***Scenario***, Menurut Munawar (2005), skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Form skenario merupakan penjelasan penulisan *use case* dari sudut pandang *actor*.

3. **Activity Diagram**, Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

2.5.2 Website

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semua nya, baik bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman web dengan halaman web lain nya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*. Website memiliki beberapa jenis berdasarkan sifat nya. Rahmat Hidayat (Gramedia, 2010:3) :

- A. Website Dinamis, merupakan sebuah website yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Bahasa pemograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan database MySQL atau MS SQL.
- B. Website Statis, merupakan website yang content nya sangat jarang diubah. Bahasa pemograman yang digunakan adalah HTML dan belum memanfaatkan database. Misalnya : web profile organisasi.

2.5.3 HTML

HTML merupakan singkatan dari Hypertext Markup Language. Disebut Hypertext karena di dalam script HTML bisa membuat agar sebuah teks menjadi

link yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan klik teks tersebut. Teks yang ber-link ini yang dinamakan hypertext karena hakikat sebuah website adalah dokumen yang mengandung banyak link untuk menghubungkan satu dokumen dengan dokumen lainnya. Jubilee Enterprise (2016:7).

2.5.4 CSS

CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. apabila kita membahasnya dalam konteks web, bisa di artikan sebagai bahasa yang digunakan untuk mengatur tampilan / desain sebuah halaman HTML. CSS = Cascading Style Sheets (Bahasa lembar Gaya). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML. Ani O. S, Ari Abdilah, Sunarti (2019:71)

2.5.5 PHP (HyperText Preprocessor)

PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script – script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side. Betha Sidik (2012 : 4).

2.5.6 Basis Data (Database)

Basis Data (database) adalah struktur data tempat menyimpan informasi yang terorganisir. Basis data terbentuk dari susunan beberapa tabel yang terdiri dari banyak baris dan kolom, Irawan (2009:37).

Database merupakan sebuah file yang mengkoordinasi file-file data yang saling berhubungan dan memiliki kepentingan yang sama sehingga akan mempermudah pengolahan data. Berikut tujuan pemanfaatan basis data Mulyanto (2009:194) :

1. Kecepatan dan kemudahan (*speed*) : agar user dapat menyimpan, memanipulasi, dan menampilkan kembali data lebih cepat dan mudah daripada cara biasa.
2. Efisiensi ruang penyimpanan (*space*) : mengurangi *redundancy*, misalnya dengan pengkodean dan membuat relasi.
3. Keakuratan (*accuracy*) : agar data sesuai dengan aturan dan batasan tertentu.
4. Ketersediaan (*availability*) : agar dapat diakses oleh setiap user yang membutuhkan.
5. Kelengkapan (*completeness*) : dengan menambahkan *field* pada tabel.
6. Keamanan (*security*) : agar data yang rahasia tidak jatuh ke tangan user yang tidak berhak, misalnya : dengan *pengkodean*, *account (username and password)*, pembedaan hak akses untuk setiap user terhadap data yang dapat dibaca atau proses yang dapat dilakukan.
7. Kebersamaan (*shareability*) : mendukung lingkungan *multiuser*, menghindari *inkonsistensi data* dan *deadlock*.

2.5.7 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multialur, multipengguna, dengan pengguna 6 juta instalasi di seluruh dunia. MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL. MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah. (Betha Sidik, 2012: 333).

2.5.8 CodeIgniter

CodeIgniter merupakan aplikasi open source yang berupa framework PHP yang menggunakan model MVC (model,view dan controller) untuk membangun sebuah website yang dinamis dengan menggunakan PHP. (David Naista, 2017:3)

CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006.

Konsep MVC merupakan konsep yang harus atau wajib diketahui terlebih dahulu sebelum mengenal framework CodeIgniter. MVC sendiri merupakan sebuah patent/teknik pemrograman yang memisahkan antara alur, data dan antarmuka suatu sistem atau bisa dikatakan secara sederhana bahwa MVC sebuah patent dalam framework yang memisahkan antara desain, data dan proses. Terdapat 3 jenis

komponen yang membangun pada MVC dalam suatu aplikasi (David Naista, 2017:7) :

1. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.

2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.

3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi. Jadi Controller bertugas sebagai penjembatani antara model yang terkoneksi dengan database dengan dibuatkan class dan functions di controllers dan dihubungkan ke views untuk ditampilkan aplikasinya.

2.6 SUBLIME TEXT

Menurut Eric Haughee dalam Sujana Cristian (2013), bahwa Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai 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 yang artinya aplikasi ini membutuhkan lisensi (license) yang harus dibeli. Akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi (license) aplikasi gratis.

