**BAB II**

**TINJAUAN PUSTAKA**

1. **Kajian Teori**
2. **Pengembangan Perangkat Lunak**

Pengembangan dalam Kamus Besar Bahasa Indonesia (KBBI) adalah suatu proses, cara ataupun perbuatan untuk mengembangkan apa yang menjadi tujuan, misalkan, pemerintah selalu berusaha dalam pengembangan pembangunan secara bertahap dan teratur yang menjurus ke sasaran yang dikehendaki.

Pengembangan *finding* tutor dapat berupa perangkat lunak atau dapat juga disebut pengembangan aplikasi. Dengan pengembangan aplikasi *finding* tutor akan berdampak pada pada siswa dengan kemudahan mencari guru les/tutor dengan kriteria yang diingikan oleh siswa.

Pengembangan perangkat lunak yang mendasari pembangunan sistem adalah *sistem development life cycle* (SDLC) atau siklus hidup pengembangan sistem. Dalam rekayasa sistem dan rekayasa perangkat lunak, SDLC adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. (Muslihuddin & Oktafianto, 2016:33). Menurut Mulyani (2016:24) mendefinisikan SDLC adalah sebuah proses logika yang digunakan oleh seseorang *system analist* untuk mengembangkan sebuah sistem informasi yang melibatkan *requirement,validation, training* dan pemilik sistem.

Konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem, yaitu proses pengembangan perangkat lunak. Tahapan proses SDLC sebagai berikut:

* 1. Tahap Perencanaan

Merupakan tahap awal dari pengembangan sistem, tahap ini bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi apa yang akan dikembangkan dan sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan serta mempertimbangkan dana yang tersedia dan siapa yang melaksanakan. Tahap perencanaan menjadi penting karena permasalahan yang sebenarnya didefinisikan dan diidentifikasi secara rinci.

* 1. Tahap Analisis

Analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem baru atau memperbaharui sistem yang sudah ada.

* 1. Tahapan Perancangan

Rancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Jika sistem ini berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan.

* 1. Tahap Penerapan

Penerapan merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Pada tahapan ini dilakukan beberapa hal yaitu: *coding, testing, instalasi*. *Output* dari tahapan ini adalah: *source code*, prosedur, pelatihan.

* 1. Tahapan Penggunaan

Tahap penggunaan terdiri dari 3 langkah, yaitu:

* + 1. Menggunakan Sistem

Pemakai menggunakan sistem untuk mencapai tujuan yang diidentifikasikan pada tahap perencanaan.

* + 1. Audit Sistem

Setelah sistem baru selesai, penelitian formal dilakukan untuk menentukan seberapa baik sistem baru itu memenuhi kriteria kinerja. Studi semacam ini disebut dengan penelaan setelah penerapan (*post implementation review*) dan dapat dilakukan oleh seseorang dari jasa informasi atau oleh auditor internal.

* + 1. Memelihara Sistem

Selama manajer menggunakan sistem, berbagai modifikasi dibuat sehingga sistem terus memberikan dukungan yang diperlukan. Modifikasi ini disebut dengan pemeliharaan sistem (*system maintenance*).

Seiring dengan perkembangan ini, metode pengembangan sistem pun mengalami pengembangan. Menurut Mulyani (2016), Adapun teknik-teknik pengembangan sistem yang merupakan pengembangan dari metode *SDLC* yaitu sebagai berikut:

1. *Prototyping*

*Prototyping* merupakan teknik pengembangan sistem yang menggunakan *prototype* untuk menggambarkan sistem, sehingga pengguna atau pemilik sistem mempunyai gambaran pengembangan sistem yang akan dilakukannya. (Mulyani, 2016). Sedangkan menurut kamus istilah informasi *prototyping* adalah strategi dalam pengembangan sistem dengan mengembangkan sistem yang lebih kecil atau bagian dari sistem dahulu pada waktu yang singkat, mengujinya dan menyempurnakannya melalui proses literasi. Dari pendapat-pendapat yang dikemukakan di atas dapat diambil suatu pengertian *prototyping* yaitu teknik pengembangan sistem yang menggunakan *prototype* untuk menggambarkan atau mewakili sistem yang akan dikembangkan.

1. *Rapid Application Development* (RAD)

McLoed dan Schel dalam Mulyani (2016) berpendapat bahwa RAD merupakan metode yang memfokuskan kecepatan dalam pengembangan sistem untuk memenuhi kebutuhan pengguna atau pemilik sistem seperti *prototyping* namun mempunyai cakupan yang lebih luas.

1. *Joint Application Development* (JAD)

JAD melibatkan pengguna atau pemilik sistem dalam melakukan pengembangan sistem, di mana ada tahap-tahap yang dilakukan dengan cara mendiskusikan sistem, dalam hal ini misalnya IT spesialis, pengguna atau pemilik sistem, sponsor dan lain sebagainya. Mulyani (2016).

1. *Unified Modeling Language* (UML)

*Unified modeling language* (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. Mulyani (2016).

1. **Les Privat**

Les privat adalah pembelajaran di luar sekolah, yang dilakukan oleh pemberi jasa di luar jam kegiatan belajar mengajar, yang menggunakan kurikulum sekolah sebagai acuan, dengan jumlah siswa dibatasi maksimal 4 orang, dan dibatasi oleh waktu tertentu. Adapun jika jumlah siswa yang diajar berkisar antara 4 sampai 9 siswa maka dinamakan les semi privat, dan lebih dari 9 siswa dinamakan les kelas klasikal.

Les privat adalah les khusus yang dilakukan kepada satu orang murid yang ditangani oleh satu orang guru/tentor. Les privat bukan hanya sekedar, “satu lawan satu”, tapi lebih dari itu, les privat adalah sebuah metode belajar mengajar dengan memperhatikan siswa sebagai seorang manusia yang unik. Ia berbeda dengan siswa lainnya, sehingga menanganinya juga harus berbeda dan harus spesial. Sebagai pengajar juga memposisikan diri secara unik/spesial.

Tujuan les privat sering untuk membantu buah hati tercinta menguasai mata pelajaran yang dianggap sulit. Karena itu, tak heran bila saat ini dapat dilihat banyaknya lembaga bimbingan belajar yang tumbuh di berbagai tempat, terutama di kota-kota. Les ini bukan pengertian konvensional tapi sesuai dengan suasana bermain dan belajar pada anak. Sementara untuk waktu belajar efektif bagi masing-masing anak ada yang siang atau sore hari. Tergantung pada keunikan masing-masing individu.

Selain berlangsung di rumah pengajar, les privat juga bisa dilakukan di rumah siswa. Karena hanya menghadapi satu murid, pengajar les memiliki waktu untuk menerangkan dengan lebih rinci, suasananya lebih santai, maka pelajaran pun biasanya lebih mudah dicerna. Yang jelas, kegiatan les ini sangat cocok bagi siswa yang berniat menekuni suatu bidang mata pelajaran tertentu.

1. **Tutor**

Tutor adalah seseorang yang memberi pelajaran atau membimbing kepada seseorang atau sejumlah kecil murid (di rumah, bukan di sekolah). Tutor sendiri memiliki makna lain yaitu dosen yang membimbing sejumlah mahasiswa dalam mata kuliah. Tutor atau guru les merupakan salah satu jalan bagi para orang tua dan juga para pelajar untuk meningkatkan kualitas akademik maupun non akademik.

Pendapat para ahli tentang pengertian tutor:

* + - 1. Tutor adalah orang yang memberi pelajaran (membimbing) kepada seseorang atau sejumlah kecil siswa dalam pelajarannya (Dedy Sugono, 2008:1022).
      2. Tutor adalah orang yang membelajarkan atau orang yang memfasilitasi proses pembelajaran di kelompok belajar (Chairudin Samosir, 2006:15).
      3. Pengertian tutor banyak dikemukakan oleh ahli pendidikan, seperti yang dikemukakan oleh Nasution (1992:4) (dalam Abi Masiku (2003:9) bahwa tutor adalah orang yang membantu murid secara individual.
      4. Hamalik (1991:73) (dalam Abi Masiku (2003:10) mengemukakan bahwa tutorial adalah bimbingan pembelajaran dalam bentuk pemberian bimbingan, bantuan, petunjuk, arahan, dan motivasi agar siswa dapat efisien dan efektif dalam belajar. Subyek atau tenaga yang memberikan bimbingan dalam kegiatan tutorial dikenal sebagai tutor. Tutor dapat berasal dari guru atau pengajar, pelatih, pejabat struktural, atau bahkan siswa yang dipilih dan ditugaskan guru untuk membantu teman-temannya dalam belajar di kelas.
      5. Pengajaran tutoring merupakan pengajaran melalui kelompok yang terdiri atas satu siswa dan satu pengajar (tutor, mentor) atau boleh jadi seorang siswa mampu memegang tugas sebagai mentor, bahkan sampai taraf tertentu dapat menjadi tutor (Winkel, 1996:401).

Tutor biasanya mendapatkan rumus cepat dari rumus-rumus umum yang sudah ada. Kemudian rumus itu diproses kembali menjadi rumus yang cepat dan tepat. Pada pengajaran soal- soal tutor memberikan resep yang dapat mengerjakan soal lebih banyak dengan waktu pengerjaan yang singkat. Pada aplikasi *Finding Tutor* ada beberapa pilihan mata pelajaran yang terdapat pada pemilihan tutor diantaranya: Matematika, Bahasa Inggris, Bahasa Indonesia, Fisika, Kimia, Biologi, mulai dari tingkat TK/PAUD, SD, SMP, SMA, Universitas hingga umum.

1. **Aplikasi**

Aplikasi komputer adalah suatu sub kelas perangkat komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Insani dan Insanuddin (2015). Menurut Juansyah (2015) Secara istilah pengertian 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. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Beberapa pengertian aplikasi tersebut dapat disimpulkan bahwa aplikasi adalah suatu perangkat lunak yang digunakan untuk melaksanakan perintah dengan tujuan tertentu oleh pengguna.

1. **REST API**

RESTful API merupakan implementasi dari API (*Application Programming Interface*). Sedangkan pengertian dari REST (*Representional State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi.

Di mana memliki tujuan untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data. selain itu *REST* lebih sering digunakan dari pada SOAP (*Simple Object Access Protocol*) karena penggunaan *REST* lebih mudah dipahami dan diterapkan dalam berbagai platform.

Ada beberapa syarat-syarat dalam arsitektur REST :

* + - * 1. *Client-Server*: Aplikasi pada umumnya dipisahkan berdasarkan *client* dan server, hal ini dimaksudkan agar pengembangan aplikasi *client* dan server dapat dilakukan secara terpisah, sehingga lebih fleksibel dan mudah di *scale*.
        2. *Stateless*: Server tidak menyimpan segala informasi tentang *client*, dengan kata lain tidak boleh ada *session*. Semua informasi tentang *client* dimiliki oleh *client*.
        3. *Cacheable* : *Cache* diperlukan untuk menghindari akses ke data *storage* untuk pengambilan informasi yang sama berulang-ulang.
        4. *Layered System*: *Client* tidak mengetahui dia terkoneksi ke server yang mana. Hal ini dibutuhkan untuk meningkatkan *scalability*.
        5. *Code On Demand*: Server dapat menambahkan/mengurangi fitur *client* dengan cepat, contohnya dengan cara mentransfer kode java *script*/java aplet.

Kemudian, RESTful API sendiri memiliki 4 komponen penting di dalamnya diantaranya adalah:

1. *URL Design*

RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti *developer*. URL API biasa disebut *endpoint* dalam pemanggilannya. Contoh penamaan URL /*endpoint* yang baik adalah seperti berikut:

*/addUser*

*/addusers/numbers*

*/addusers/numbers/address*

*/addusers/numbers/address/abc*

1. *HTTP Verbs*

Setiap *request* yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di *request client*, diantaranya yang umum dipakai adalah:

1. *GET*

*GET* adalah metode HTTP *Request* yang paling simpel, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

Contoh:

*GET* /users: Mengembalikan daftar *user*

*GET* /users/1234 : Mengembalikan data *user* dengan ID 1234

1. *POST*

*POST* adalah metode *HTTP Request* yang digunakan untuk membuat data baru dengan menyisipkan data dalam *body* saat *request* dilakukan.

Contoh:

*POST* /users: Membuat data *user* baru

1. *PUT*

*PUT* adalah metode *HTTPRequest* yang biasanya digunakan untuk melakukan *update* data *resource*.

Contoh:

*PUT* /users/1234: Mengupdate data *user* dengan ID 1234

1. *DELETE*

*DELETE* adalah metode HTTP *Request* yang digunakan untuk menghapus suatu data pada *resource.*

Contoh:

*DELETE* /users/1234: Menghapus data *user* dengan ID 1234

Selain HTTP *Verbs* di atas, masih ada metode *HEAD* dan *PATCH* dalam HTTP *Request*, tetapi 4 metode *request* di atas tersebut adalah yang paling umum digunakan.

1. HTTP *Response Code*

HTTP *response code* adalah kode standarisasi dalam menginformasikan hasil *request* kepada *client*. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu:

2XX: adalah *response code* yang menampilkan bahwa *request* berhasil.

4XX: adalah *response code* yang menampilkan bahwa *request* mengalami kesalahan pada sisi *client*.

5XX: adalah *response code* yang menampilkan bahwa *request* mengalami kesalahan pada sisi server.

1. *Format Response*

Setiap *request* yang dilakukan *client* akan menerima data *response* dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data *response* tersebut barulah *client* bisa menggunakannya dengan cara memparsing data tersebut dan diolah sesuai kebutuhan.

1. ***Android***

*Android* adalah sebuah sistem operasi berbasis *linux* yang bersifat terbuka (*open source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. (Salbino, 2014:7). *Android* merupakan sistem operasi *open source* yang di mana semua orang bisa mengembangkannya, hal itulah yang membuat perkembangan aplikasi *android* semakin cepat dan berkembang. (Masruri, 2015: 1)

*Android* awalnya dikembangkan oleh *android, inc*, dengan dukungan finansial dari *google* yang kemudian membelinya pada tahun 2015. Sistem operasi ini dirilis secara resmi pada tahun 2007. Akan tetapi terkhusus pada *smartphone* yang pertama kali meggunakan sistem operasi berbasis *android* pertama kali dijual pada bulan oktober 2018. Fuad (dalam Nasrullah, 2018).

Berikut adalah tabel yang memuat berbagai macam versi sistem operasi *android* mulai dari versi pertama hingga versi terbaru:

Tabel 2.1

Jenis-jenis Versi Sistem *Android*

|  |  |  |
| --- | --- | --- |
| Nama | Versi | Tahun Rilis |
| *Android* Beta | Beta | 5 November 2007 |
| *Android* | 1.0 | 23 September 2008 |
| *Cupcake* | 1.5 | 23 September 2008 |
| *Donut* | 1.6 | 30 April 2009 |
| *Éclair* | 2.0 – 2.1 | 15 September 2009 |
| *Froyo* | 2.2 – 2.2.3 | 26 Oktober 2009 |
| *Gingerbread* | 2.3 – 2.3.7 | 20 Mei 2010 |
| *Honetcomb* | 3.0 – 3.2.6 | 6 Desember 2010 |
| *Ice Cream Sandwich* | 4.0 – 4.0.4 | 22 Februaru 2011 |
| *Jelly Bean* | 4.1 – 2.3.1 | 18 Oktober 2011 |
| *Kitkat* | 4.4 – 4.4.4 | 9 Juli 2012 |
| *Lollipop* | 5.0 – 5.1.1 | 31 Oktober 2013 |
| *Marshmellow* | 6.0 – 6.0.1 | 5 Oktober 2014 |
| *Nougat* | 7.12(Beta) | 30 Januari 2017 |
| *Oreo* | 8.0 | 21 Agustus 2017 |
| *Pie* | 9.0 | 6 Agustus 2018 |

Sumber: www.tekno.kompas.com

1. **Pengujian Perangkat Lunak**

Menurut Liana (2015). Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Pengujian terhadap perangkat lunak dilakukan untuk menghindari banyaknya kesalahan. Pengujian perangkat lunak tidak hanya dilakukan untuk meminimalisir kesalahan secara teknik tetapi juga kesalahan non teknis. Kesalahan non teknis tersebut seperti pengujian pesan kesalahan sehingga user tidak bingung.

Model ISO-25010 merupakan bagian dari *Software Product Quality Requirements and Evaluation* (SQuaRE), di mana model ini berkaitan dengan model kualitas perangkat lunak yang merupakan pengembangan dari model sebelumnya yaitu model ISO-9126. Pada model ini terdapat beberapa sub-karakteristik tambahan dan beberapa sub-karakteristik yang dipindahkan ke karakteristik lain. ISO-25010 mengidentifikasi ada delapan karakteristik kualitas perangkat lunak, yaitu: *functionality Suitability, performance efficiency, compability, usability, reliability, security, maintainability* dan *portability.*

Berikut ini merupakan pengertian dari masing-masing faktor dan sub-faktor yang terdapat pada model ISO-25010, antara lain:

* + - * 1. *Functionality Suitability*

Karakteristik ini menunjukan sejauh mana suatu produk atau sistem menydiakan fungsi yang memenuhi kebutuhan ketika digunakan dalam kondisi tertentu*. Functionality suitability* mempunyai beberapa subkarakter yaitu:

* 1. *Functional completenesss* adalah sejauh mana serangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
  2. *Functional correctness* adalah sejauh mana suatu produk atau sistem membarikan hasil yang benar dan tingkat presisi yang dibutuhkan.
  3. *Functional appropriatenes* sejauh mana fungsi memfasilitasi penyelesaian tugas dan tujuan yang ditentukan.
     + - 1. *Performance Efficiency*

Karakteristik ini mewakili kinerja *relative* terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. *Performance efficiency* mempunyai beberapa sub karakteristik yaitu:

1. *Time behavior* merupakan tingkat di mana perangkat lunak dapat memberikan reaksi pada waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi pada kondisi tertentu.
2. *Resource utilization* merupakan tingkat di mana perangkat lunak menggunakan beberapa sumber daya ketika perangkat lunak melakukan aksi dari sebuah fungsi pada kondisi tertentu.
3. *Copacity* merupakan sejauh mana batas maksimum suatu produk atau parameter sistem memenuhi persyaratan.
   * + - 1. Compatibility

*Compability* merupakan sejauh mana produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen lain dan menjalankan fungsi yang disyaratkan, sambil berbagi lingkungan perangkat keras atau perangkat lunak yang sama. *Performance efficiency* mempunyai beberapa sub karakteristik yaitu:

1. *Co-existence* merupakan tingkat sejauh mana suatu produk dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak yang merugikan pada produk lain.
2. *Interoperability* merupakan tingkat Sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.
   * + - 1. *Usability*

*Usability* merupakan tingkat sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan. *Portability* mempunyai beberapa sub-karakteristik yaitu:

1. *Appropriateness recognisability* merupakan tingkat sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.
2. *Learnability* merupakan tingkat di mana perangkat lunak mudah untuk dipelajari penggunaannya oleh para pengguna.
3. *Operability* merupakan tingkat di mana produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan oleh pengguna.
4. *User error protection* merupakan tingkat di mana sistem melindungi pengguna dari membuat kesalahan.
5. *User interface aestetics* merupakan tangkat di mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
6. *Accessibity* merupakan tingkat di mana sistem dapat digunakan oleh orang-orang dengan jangkauan karakteristik dan kemampuan terluas untuk mencapai tujuan yang ditentukan dalam kontek penggunaan yang ditentukan.
   * + - 1. *Reliability*

*Reliabilty* merupakan sejauh mana sistem, produk atau komponen melakukan fungsi yang ditentukan dalam kondisi yang ditentukan untuk periode waktu tertentu. *Reliability* mempunyai beberapa sub karakteristik yaitu:

1. *Matury* merupakan tingkat di mana suatu sistem, produk atau komponen memenuhi kebutuhan untuk keandalan dalam operasi normal.
2. *Availability* merupakan tingkat di mana sistem, produk atau komponen beroperasi dan dapat diakses bila diperlukan untuk digunakan.
3. *Fault tolerance* merupakan tingkatan di mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun ada kesalahan perangkat keras atau perangkat lunak.
4. *Recoverability* merupakan tingkat di mana dalam hal terjadi gangguan atau kegagalan, suatu produk atau sistem dapat dipulihkan data yang terkena dampak langsung dan membangun kembali keadaan sistem yang diinginkan.
   1. *Security*

Faktor ini merupakan tingkat di mana suatu produk atau sistem melindungi informasi dan data sehingga orang, produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasi mereka. *Security* mempunyai beberapa subkarakteristik yaitu:

1. *Confidentiality* merupakan tingkat di mana suatu produk atau suatu sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
2. *Integrity* merupakan tingkat di mana suatu sistem mencegah akses yang tidak sah diakses oleh mereka atau modifikasi data.
3. *Non-repudiation* merupakan tingkat di mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tersebut tidak dapat ditolak kemudian.
4. *Accountability* merupakan tingkat di mana tindakan suatu entitas dapat ditelusuri.
5. *Authenticity* merupakan tingkat di mana identitas subjek atau sumber daya dapat dibuktikan sebagai yang di klaim.
   1. *Maintainability*

*Maintainability* merupakan tingkat di mana sebuah perangkat lunak dapat dimodifikasi. Modifikasi ini termasuk perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik. *Maintainability* mempunyai beberapa subkarakteristik yaitu:

1. *Modularity* merupakan tingkatan di mana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain.
2. *Reusability* merupakan tingkatan di mana asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan *asset* lainnya.
3. *Analyzability* merupakan tangkat di mana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau mengidentifikasi bagian yang dapat dimodifikasi.
4. *Modifability* merupakan tingkat di mana suatu produk atau sistem dapat secara efektif dan efisien dimodifikasi tanpa memperkenalkan cacat atau menurunkan kualitas produk yang ada.
5. *Testability* merupakan tingkat efektivitas dan efisiensi yang dengannya kriteria pengujian dapat ditetapkan untuk suatu sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.
   1. *Portability*

Merupakan tingkat di mana perangkat lunak dapat berpindah dari lingkungan yang satu kelingkungan yang lain.

1. *Adaptability* merupakan kemudahan di mana sistem atau komponen dapat berpindah dari lingkungan *hardware* atau perangkat satu ke *hardware* atau perangkat lunak lainnya.
2. *Installability* merupakan tingkat efektivitas dan efisiensi yang dengannya suatu produk atau sistem dapat berhasil diinstal dan dihapus dalam lingkungan tertentu.
3. *Replaceabilty* merupakan tingkat di mana suatu produk dapat menggantikan produk perangkat lunak tertentu lainnya untuk tujuan yang sama di lingkungan yang sama.(ISO/IEC, 2011)
4. **KAJIAN YANG RELEVAN**

Dalam melakukan penelitian penulis mengambil referensi dari kajian penelitian yang memiliki relevansi variabel yang hampir sama. Sebagai bahan acuan serta pembelajaran dalam menyusun penelitian ini sebagai berikut:

* + - 1. Riska Adhita, (2017) Penelitian yang bertujuan membangun sebuah aplikasi mobile berbasis android untuk menemukan guru privat/les dan penentua prioritas seleksi murid. Aplikasi ini menggunakan perangkat bergerak yang dapat menunjang mobilitas para murid, serta para tutor atau guru les. Pada aplikasi ini akan menggunakan *Google* API untuk menunjang pencarian tutor dan mendapatkan prioritas murid. Pembangunan aplikasi *Finding* Tutor daerah Surabaya ini dibangun dengan bahasa JSON atau singkatan dari Java Script Object Notation, yaitu sebuah format untuk pertukaran data. Aplikasi berbasis *mobile android* yang mengimplementasikan semua kebutuhan mulai dari penyediaan informasi tutor atau guru les yang ada, pencarian tutor atau guru les sesuai dengan kriteria yang diinginkan, serta pencocokan yang berujung pada pembuatan prioritas bagi tutor atau guru les untuk mendapatkan murid yang sesuai dengan kriteria yang diinginkan.
      2. Aplikasi Reservasi Les *Private* di *Easys peak* Denpasar Berbasis Web Dan *Android* oleh Fery Yudhitama Putra, dkk pada tahun 2016. Hasil penelitian ini terdiri dari web dan *android*. Pembangunan aplikasi reservasi les *private* dibangun dengan bahasa pemrograman PHP dan Java dengan menggunakan 30 *framework Codeigniter* pada sisi web, sedangkan pada sisi *android* menggunakan *tool Eclipse*, serta MySQL sebagai media penyimpanan basis data. Aplikasi reservasi les privat ini memiliki beberapa fungsi untuk melakukan pemesanan waktu dan tutor yang dapat dilakukan sendiri oleh *student* dari *Easyspeak* dan pada sisi tutor aplikasi dapat memberikan informasi student yang akan diajar, serta pada sisi operator dapat memberikan kemudahan dalam melakukan pengaturan *booking* les *private* karena sudah terkomputerisasi tidak manual seperti dulu. Aplikasi reservasi les privat juga dilengkapi dengan sistem *reminder* atau pengingat yang dibuat pada sisi aplikasi android, dengan menggunakan sistem *alarm* manager.
      3. Sistem Reservasi Tempat *Meeting* Berbasis *Android* oleh Desta Aditya Kusumajati pada tahun 2017. Hasil penelitian ini adalah suatu sistem informasi reservasi tempat *meeting* berbasis *android* untuk *user*/pengguna dan berbasis *website* untuk admin di lokasi tempat *meeting*. Fitur yang diberikan pengguna adalah pengguna dapat memesan tempat meeting berdasar waktu, tempat yang diinginkan dan jumlah kapasitas peserta *meeting* yang bisa dilakukan tanpa pengguna harus reservasi langsung kelokasi dan mendapatkan informasi pembayaran setelah melakukan pemesanan. Untuk admin diberi *website* admin yang digunakan untuk memantau pesanan yang masuk. Dengan dilakukannya penelitian ini dapat mempermudah sejumlah orang atau instansi dan perusahaan yang ingin memesan tempat *meeting* secara cepat disaat waktu yang tidak terduga atau dari jauh jauh hari tanpa harus *review* lokasi *meeting* dengan cara 31 mengunjungi satu persatu lokasi tetapi hanya dengan melihat *review* tempat melalui aplikasi. Mereka langsung dapat menentukan tempat seperti apa yang diinginkan serta menentukan ruangan yang memadahi dengan jumlah peserta meeting.

1. **Kerangka Pikir**

Penelitian ini dimulai dengan adanya observasi permasalahan, sehingga diperlukan alternatif penyelesaian masalah. Penyelesaian masalah dengan membuat aplikasi reservasi pencarian tutor berbasis Android. Adapun kerangka pikir dalam penelitian ini adalah:

Pengembangan aplikasi pencarian tutor

Berbasis *Android*

Pemanfaatan teknologi di *android mobile* sebagai pengembangan sistem informasi pencarian tutor

Pengembangan aplikasi pencarian tutor berbasis android

1. Mengembangkan aplikasi *Finding* Tutor
2. Mengetahui hasil pengujian aplikasi *Finding* Tutor menggunakan ISO 25010
3. Mengetahui tanggapan pengguna terhadap pengembangan aplikasi *Finding* Tutor

Gambar 2.1 Kerangka pikir