



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Sebelumnya

Pada penelitian sebelumnya yang berjudul *Gamification-based e-learning Platform for Computer Programming Education*, penelitian ini menggunakan konsep *gamification* yaitu dalam sebuah *platform* penunjang pengajaran dalam pemrograman komputer, pada penelitian ini pengajar mendistribusikan atau memberikan materi-materi yang nantinya harus dikerjakan siswa. Pada penelitian ini pengajar bisa melihat sejauh mana siswa telah mengerjakan materi serta melihat komunikasi antar siswa di dalam group belajar (Swacha et al., 2013).

Pada penelitian yang berjudul *A game plan: Gamification Design Principles in mHealth Applications For Chronic Disease Management*, pada penelitian ini menjelaskan tentang sebuah konsep manajemen diri pasien dalam pengelolaan penyakit kronis yang efektif dan sangat penting untuk memperbaiki hasil kesehatan yang positif, dan strategi insentif yang dikemas dalam aplikasi yang menarik pengguna/pasien (Miller et al., 2014).

Pada penelitian yang berjudul *Penerapan Konsep Gamification Pada Pembelajaran Tenses Bahasa Inggris Berbasis Web*. Pada penelitian ini menggunakan konsep *Gamification* yang membahas serta mempelajari 12 (dua belas) tenses di dalam bahasa inggris dengan menarik (Al Fatta et al., 2015).

#### 2.2 TOEIC (*Test of English for International Communication*)

TOEIC merupakan salah satu jenis tes yang dikembangkan oleh ETS (*Educational Test Service*), tes ini akan mengukur kemampuan seseorang untuk berkomunikasi dalam Bahasa inggris dalam konteks kehidupan sehari-hari dan lingkungan kerja secara global dengan menggunakan ungkapan yang digunakan dan kosa kata yang digunakan sehari-hari (Haira et al., 2016).



Dalam tes TOEIC terdapat 200 pertanyaan yang terdiri dari *Listening Comprehension* dan *Reading Comprehension*. Semua pertanyaan memiliki 4 (empat) pilihan ganda sebagai pilihan jawaban. Untuk bagian *listening* terdiri dari *photograph*, *question-response*, *short conversation*, dan *talk*. Sedangkan untuk *reading* terdiri dari *incomplete sentence*, *text completion*, *single passages*, dan *double passage*.

### 2.2.1 Bagian - bagian TOEIC

1. Listening Comprehension
  - a. Photographs (10 pertanyaan)
  - b. Question-response (30 pertanyaan)
  - c. Conversation (30 pertanyaan)
  - d. Talks (30 pertanyaan)
2. Reading Comprehension
  - a. Incomplete sentence (40 pertanyaan)
  - b. Text completion (12 pertanyaan)
  - c. Single passage (28 pertanyaan)
  - d. Double passage (20 pertanyaan)

### 2.2.2 Skor TOEIC

Seperti yang telah dijelaskan diatas bahwa TOEIC terdiri dari 200 pertanyaan yang terdiri dari *Listening Comprehension* dan *Reading Comprehension* dan untuk waktu pengerjaannya yaitu 120 menit atau 2 jam. Dua jenis kemampuan yang diujikan tersebut akan memiliki nilai terpisah antara 5 sampai 495. Sehingga, untuk memperoleh hasil tes dapat didapatkan dengan menambahkan nilai yang ada. Sehingga, total nilai TOEIC memiliki skala antara 10 sampai 990.

Jumlah skor yang telah dikumpulkan setelah menyelesaikan TOEIC akan menunjukkan kemampuan anda. Di dalam TOEIC, terdapat beberapa tingkatan kemampuan Bahasa Inggris seseorang yang ditunjukkan dari tabel berikut ini:



| Skor      | Level                           |
|-----------|---------------------------------|
| 10 - 250  | Novice                          |
| 255 - 400 | Elementary                      |
| 405 - 600 | Intermediate                    |
| 605 - 780 | Basic working proficiency       |
| 785 - 900 | Advanced working proficiency    |
| 905 - 990 | General profesional proficiency |

Tabel 2.1 Tabel skor dan level TOEIC

## Keterangan:

1. *Novice*

Pada tingkatan ini, seseorang:

- Membaca memo dan menu sederhana, jadwal bis dan kereta api, rambu lalu lintas, dan lain-lain.
- Memahami pertanyaan sederhana dalam situasi sosial.

2. *Elementary*

Pada tingkatan ini, seseorang:

- Memahami pesan singkat melalui telepon
- Memahami pengenalan sederhana
- Memahami korespondensi bisnis yang sederhana

3. *Intermediate*

Pada tingkatan ini, seseorang:

- Memahami hal-hal yang berkaitan dengan pekerjaan rutin
- Memahami pengumuman perjalanan
- Percakapan sosial yang terbatas
- Memahami tata cara penggunaan untuk pemula
- Menggunakan kamus untuk memahami dokumen teknis yang lebih tinggi
- Membaca agenda untuk rapat



#### 4. *Basic working proficiency*

Pada tingkatan ini, seseorang:

- Memahami penjelasan tentang masalah pekerjaan
- Meminta produk melalui telepon
- Memahami diskusi tentang hal-hal yang terjadi pada saat itu oleh penutur bahasa inggris.
- Memahami berita di radio

#### 5. *Advanced working proficiency*

Pada tingkatan ini, seseorang:

- Memahami pembicaraan dalam bahasa inggris pada rapat internasional
- Membaca hampir semua jenis dokumen dengan berbagai tingkatan kesulitan
- Membaca subjek tertentu dengan tingkatan teknis tinggi dengan sedikit bantuan kamus

#### 6. *General professional proficiency*

Pada tingkatan ini, seseorang:

- Memahami penutu bahasa inggris di dalam suatu rapat
- Dapat membaca dengan baik untuk kebutuhan dunia professional

### 2.2.3 Cakupan Soal TOEIC

Berikut ini adalah beberapa bidang yang muncul pada tes TOEIC:

- a. Perkembangan perusahaan (perkembangan produk, riset)
- b. Acara makan (makan siang dengan kolega, jamuan makan, resepsi, reservasi, restoran)
- c. Hiburan (bioskop, musik, seni, museum, media, pameran)
- d. Keuangan (bank, investasi, pajak, akuntansi, penagihan)
- e. Bisnis (kontrak, negosiasi, merger, penjualan, pemasaran, asuransi, rencana bisnis, konferensi)
- f. Kesehatan (asuransi kesehatan, mengunjungi dokter, dokter gigi, klinik, rumah sakit)



- 
- g. Perumahan/perusahaan properti (kontruksi, pembelian dan penyewaan, layanan listrik dan gas)
  - h. Manufaktur (manajemen perusahaan, *quality control*)
  - i. Kantor (rapat direksi, komite, surat-menyurat, telepon, fax, email, peralatan kantor, perabot, peraturan kantor)
  - j. HRD (perekrutan, pengunduran diri, gaji, promosi, lamaran pekerjaan, iklan pekerjaan, pension, penghargaan)
  - k. Pembelian (pembelajaan, pengiriman, faktur)
  - l. Wilayah teknis (elektronik, teknologi, computer, labolatorium, spesifikasi teknis)
  - m. Perjalanan (kereta api, pesawat, taksi, bis, kapal feri, tiket, jadwal, pengumuman di stasiun dan bandara, penyewaan mobil, hotel, reservasi, penundaan dan pembatalan tiket).

### 2.3 *Gamification*

*Gamification* yaitu menerapkan mekanisme game untuk aktivitas non game untuk mengubah perilaku orang (Bunchball, 2010).

*Gamification* merupakan penggunaan unsur-unsur permainan kedalam konteks yang tidak berhubungan dengan permainan (contoh: pendidikan, kesehatan, pemasaran, dan lain sebagainya) dengan tujuan menarik perhatian dan motivasi pengguna (Zichermann et al., 2012).

*Gamification* adalah penggunaan metafora desain *game* untuk menciptakan lebih seperti game dan pengalaman menarik (Marczewski, 2015).

Berikut ini unsur-unsur permainan yang umumnya diterapkan pada bidang yang akan digamifikasikan antara lain:

#### a. *Badges* / Lencana

Lencana, medali, piala, dan konsep-konsep lain digunakan untuk mengidentifikasi dan menghargai prestasi individu pengguna. Sebagian pengguna menikmati kejutan mendadak atau kesenangan ketika lencana tak terduga muncul



dalam permainan. Hal ini diharapkan dapat menarik minat pengguna agar tertarik untuk terus bermain (Zichermann et al., 2012).

b. *Leaderboards* / Papan Peringkat

Papan peringkat secara dinamis menunjukkan perkembangan peringkat pengguna dibandingkan dengan pengguna yang lain. Dengan papan peringkat memberikan pandangan pada pengguna akan seberapa baik ia menggunakan aplikasi dibandingkan dengan pengguna lainnya (Alexandrov et al., 2011).

c. Poin dan level

Poin dan sistem level dilaksanakan untuk memberitahukan perkembangan pengguna dari tingkat awal dilanjutkan keahlian dan pengetahuan. Hal ini juga untuk mengetahui keterlibatan pengguna, sebagai pengguna baru maupun yang sudah berpengalaman dibuktikan melalui besarnya poin dan level yang mereka miliki (Zichermann et al., 2012).

d. Relasi

Hubungan yang dialami ketika pengguna terhubung dengan pengguna lain dengan memungkinkan berbagi data dan informasi antara melalui platform media sosial, seperti Facebook dan Twitter. Integrasi antara sistem dan jaringan sosial sangat menarik untuk memperkuat motivasi terhadap pengguna (A.F. Aparicio, 2012).

e. *Achievements* / *Rewards*

Sebuah hadiah berwujud atau tidak berwujud, disajikan setelah terjadinya suatu tindakan (yaitu, perilaku) dengan maksud untuk menyebabkan perilaku terjadi lagi (Marczewski, 2015).

### 2.3.1 Deskripsi Metode *Gamification* yang Digunakan

Pada penelitian yang telah dilakukan oleh Swacha et al., 2013, Miller et al., 2014, dan Al Fatta et al., 2015, pada dasarnya mereka menggunakan konsep *Gamification* yang melibatkan kegiatan antara lain:



a. Identifikasi pokok pembahasan

Identifikasi tujuan utama dari pokok masalah. Setiap tugas atau pekerjaan yang ingin dibahas terdiri dari tujuan utama yang dapat diidentifikasi dengan jelas.

b. Identifikasi tujuan transversal

Identifikasi satu atau lebih tujuan yang menarik minat pengguna. Berdasarkan hal itu kita akan membuat sistem yang didasarkan pada permainan mekanik yang meningkatkan motivasi dan kepentingan pengguna.

c. Pemilihan unsur-unsur permainan

Pemilihan unsur-unsur permainan sesuai dengan mekanisme dengan tujuan dan mendukung kebutuhan untuk memotivasi pengguna. Beberapa unsur permainan terbagi menjadi beberapa mekanisme antara lain berupa:

- i. Kemandirian, merujuk kepada kesadaran diri individu untuk melakukan suatu kegiatan berdasarkan kepentingan pribadi. Teknik-teknik untuk meningkatkan kemandirian adalah dengan menggunakan umpan balik positif dan memberikan pengguna kebebasan untuk melaksanakan instruksi yang diberikan. Unsur permainan yang terlibat pada mekanisme ini yaitu profil, avatar, konfigurasi antarmuka, alternatif kegiatan, kontrol privasi, kontrol pemberitahuan.
- ii. Kompetensi, adalah kebutuhan individu terhadap kemampuan dalam menyelesaikan berbagai tantangan yang diberikan. Meningkatkan persaingan dengan memperoleh pengetahuan atau keterampilan baru, menerima umpan balik positif atau mendapatkan tantangan bisa meningkatkan motivasi. Unsur permainan yang terlibat pada mekanisme ini yaitu umpan balik positif, tantangan yang optimal, informasi progresif, *badges*, *points*, *levels*, *leader boards*.

d. Analisa keefektifan: Analisis efektivitas penerapan *Gamification* berdasarkan faktor kesenangan, indikator kualitas dan kepuasan, serta kualitas layanan. Ini akan dilakukan melalui uji pengguna dengan cara penyelesaian kuesioner dan tes pengguna dengan menerapkan aturan-aturan menentukan batas menang dan kalah. Aturan - aturan dalam game ini disebut juga dengan *game rule*.



## 2.4 Pengujian *Software*

Pengujian *software* sangat di perlukan untuk memastikan apakah *software* yang sedang atau sudah di buat dapat berjalan sesuai fungsionalitas yang diharapkan. Dengan adanya pengujian *software* kita dapat mendeteksi kesalahan ataupun kekurangan pada *software* yang kita buat agar secepatnya bisa dieteksi dan koreksi secepatnya. Pengujian atau testing merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan *software* sepeerti halnya analisis, desain, dan pengkodean (Shi, 2010).

Dalam melakukan pengujian *software* haruslah mempunyai karakteristik, langkah-langkah pengujian, dan pengimplementasian strategi *testing software* seperti yang dijelaskan oleh (Baht et al., 2015).

Karakteristik pengujian:

- a. *Testing* dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasiskan komputer.
- b. Teknik *testing* yang berbeda sesuai dengan poin - poin yang berbeda pada waktunya.
- c. Testing diadakan oleh pembuat atau pengembang *software* dan untuk proyek yang besar oleh *group independent*.
- d. *Testing* dan *Debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomondasikan pada setiap strategi *testing*.

Langkah-langkah pengujian *software* ada 4 (empat) yaitu:

- a. Unit *testing-testing* per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian *error* secara maksimum.
- b. Integration *testing-testing* perpenggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan kontruksi program.
- c. *High order test* yaitu terjadi ketika *software* telah selesai diintegrasikan atau dibangun menjadi satu tidak terpisah-pisah.



- d. *Validation test* yaitu menyediakan jaminan akhir bahwa *software* memenuhi semua kebutuhan fungsional, kepribadian dan performa.

Selain itu Gilb menyatakan bahwa prosedur yang harus digunakan jika ingin mengimplementasikan strategi *testing* yang sukses yaitu:

- a. Menetapkan seluruh kebutuhan produk *software* dalam perhitungan sebelum memulai *testing*.
- b. Status obyek *testing* harus jelas.
- c. Memahami pengguna *software* dan mengembangkan sebuah profil untuk setiap kategori *user*.
- d. Mengembangkan rencana *testing* yang menekankan pada “*rapid cycle testing*”.
- e. Membangun *software* yang sempurna yang didesain untuk menguji dirinya sendiri.
- f. Menggunakan tinjauan ulang yang formal sebagai *filter* sebelum pengujian
- g. Melakukan tinjauan ulang secara formal untuk menilai strategi tes dan kasus tes itu sendiri.
- h. Mengembangkan pendekatan peningkatan yang berkelanjutan untuk proses *testing*.

Ada beberapa jenis pengujian *software* (perangkat lunak), antara lain (Khan, 2011):

- a. *White Box Testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.
- b. *Black Box Testing* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Pada pengujian *software* ini penulis melakukan pengujian dengan metode *Black Box Testing*.



### 2.4.1 *Black Box Testing*

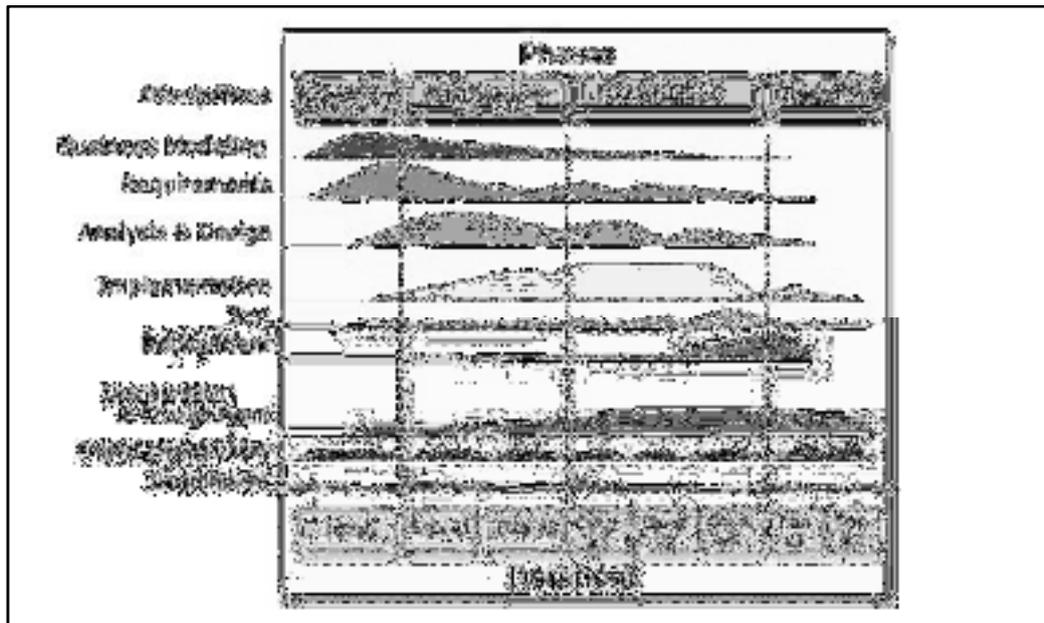
*Black box testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program.

*Black box testing* cenderung untuk menemukan hal-hal berikut:

- a. Fungsi yang tidak benar atau tidak ada.
- b. Kesalahan antarmuka (*interface errors*).
- c. Kesalahan pada struktur data dan akses baris data.
- d. Kesalahan performansi (*performance errors*)
- e. Kesalahan inisialisasi dan terminasi.

## 2.5 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang akan digunakan adalah *Rational Unified Process* (RUP). RUP merupakan suatu pendekatan yang mengatur tugas-tugas dan tanggung jawab pada organisasi pengembang perangkat lunak (Kruchten, 2000). RUP ditujukan untuk menjamin kualitas perangkat lunak yang akan dibangun. RUP mendukung pembangunan perangkat lunak yang dapat direkayasa secara iteratif, mengatur kebutuhan perangkat lunak, memodelkan perangkat lunak secara visual, dan secara berkelanjutan menjaga kualitas perangkat lunak yang akan dibangun. Fase-fase dalam RUP dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Arsitektur *Rational Unified Process*

Sumber: (Kruchten, 2000)

Pada Gambar 2.1 ditunjukkan Arsitektur dari RUP dimana terdapat 4 Fase dalam proses pengembangan sistem, yaitu Inception, Elaboration, Construction, dan Transition.

Dari keempat fase ini, pada setiap fase dilakukan berbagai aktifitas yaitu Pemodelan Bisnis, Pengumpulan Kebutuhan, Analisis dan Desain, Implementasi, *Testing*, Penyebarluasan, Konfigurasi dan Pengelolaan perubahan, serta pengelolaan lingkungan proyek. Pada setiap Fase, aktifitas yang dilakukan memiliki porsi yang berbeda. Seperti pada Fase Inception, akan dilakukan lebih banyak aktifitas pemodelan dan pengumpulan kebutuhan dan telah dilakukan sedikit Analisis dan Desain, Implementas, serta *Testing*. Namun aktifitas Penyebarluasan belum dilakukan sama sekali.

Fase-fase dalam RUP antara lain (Kruchten, 2000):

1. *Inception*

Tahap *inception* merupakan tahap awal dari fase pengembangan RUP. Tujuan utama dari fase *inception* ini antara lain menetapkan lingkup proyek perangkat lunak dan kondisi batas, membedakan kasus penggunaan sistem, memperkirakan biaya keseluruhan dan jadwal untuk seluruh



proyek, memperkirakan risiko potensial (sumber-sumber yang tidak bisa diprediksi), dan mempersiapkan lingkungan pendukung untuk proyek.

Pada fase ini, kegiatan yang dilakukan adalah:

- i. Menentukan spesifikasi proyek.
- ii. Merencanakan dan mempersiapkan *business case*, mengevaluasi alternatif untuk rencana proyek, dan penjadwalan proyek.
- iii. Mempersiapkan lingkungan untuk proyek yaitu dengan menaksir proyek, memilih *tools* yang akan digunakan, dan memutuskan bagian dari proses mana yang akan ditingkatkan.

## 2. *Elaboration*

Tahap *elaboration* merupakan tahap kedua pada RUP. Tujuan dari fase *elaboration* adalah untuk menganalisis permasalahan, menentukan arsitektur dasar, membangun rencana proyek, dan mengeliminasi elemen-elemen resiko terbesar dari proyek.

Kegiatan yang dilakukan pada fase ini adalah:

- i. Menggambarkan garis awal arsitektur secara praktis.
- ii. Menyempurnakan visi berdasarkan informasi baru yang diperoleh selama fase *elaboration* dengan membentuk pemahaman yang kuat dari permasalahan untuk arsitektural dan perencanaan keputusan.
- iii. Membuat dan menggambarkan garis besar rencana iterasi lebih detail untuk fase *construction*.
- iv. Menyempurnakan *business case* dan meletakkan pada tempatnya lingkungan pengembangan, termasuk proses, *tools* dan pendukung diperlukan untuk fase *construction*.
- v. Menyempurnakan arsitektur dan memilih komponen.

## 3. *Construction*

Fase *construction* merupakan fase ketiga yang merupakan tahap untuk mengimplementasikan hasil desain ke *coding* dan melakukan pengujian hasil implementasi. Pada fase ini dilakukan sederetan iterasi. Pada setiap iterasi akan melibatkan proses analisis, desain, implementasi dan *testing*. Tujuan utama dari fase *construction* adalah meminimalkan



biaya pengembangan dengan mengoptimalkan sumber daya dan menghindari hal yang tidak perlu dan pengerjaan ulang, mencapai kualitas yang memadai, dan melengkapi analisis, desain, pengembangan dan pengujian dari semua fungsi yang diperlukan.

Pada fase ini, kegiatan yang dilakukan adalah:

- i. Pengaturan sumber daya, kontrol, dan pengoptimalan proses.
- ii. Melengkapi komponen pengembangan dan pengujian terhadap evaluasi kriteria.
- iii. Penilaian dari hasil akhir produk.

#### 4. *Transition*

Tahap *transition* merupakan tahap akhir dari metode pengembangan RUP. Fase ini terfokus untuk memastikan bahwa produk layak untuk pengguna akhir. Pada fase ini terdapat beberapa iterasi, termasuk ujian produk dalam mempersiapkan produk akhir, dan membuat penyesuaian berdasarkan *feedback* pengguna. Pada fase ini dilakukan *Beta testing* untuk memvalidasi sistem baru terhadap harapan pengguna.

*Rational Unified Process* memiliki langkah praktis yang terbaik di dalam pengembangan perangkat lunak, yaitu (Kruchten, 2000):

##### a. Mengembangkan perangkat lunak secara iteratif

Pendekatan yang dilakukan secara iteratif dapat meningkatkan pemahaman tentang masalah yang ada seiring dengan dilakukannya perbaikan sehingga secara bertahap dapat menemukan solusi yang efektif atas beberapa iterasi.

##### b. Mengelola kebutuhan

Merupakan pendekatan sistematis untuk memperoleh, mengatur, berkomunikasi, dan mengontrol perubahan kebutuhan dari sistem perangkat lunak intensif atau aplikasi. Keuntungan dari pengelolaan kebutuhan yaitu:

- i. Pengontrolan yang lebih baik terhadap proyek yang kompleks.
- ii. Meningkatkan kualitas perangkat lunak dan kepuasan pengguna.
- iii. Mengurangi biaya produksi dan penundaan.



- iv. Meningkatkan komunikasi tim.
- c. Menggunakan *component-based* arsitektur

RUP menyediakan sebuah metode pendekatan secara sistematis untuk mendesain, membangun, dan mendefinisikan sebuah arsitektur dengan menggunakan komponen baru maupun komponen yang ada.
- d. Pemodelan perangkat lunak secara visual

Model membantu kita untuk memahami dan membentuk baik masalah maupun solusinya. Pemodelan dapat dilakukan dengan menggunakan UML (*Unified Modelling Language*).
- e. Memverifikasi mutu perangkat lunak secara terus-menerus

Dalam pengembangan perangkat lunak, kualitas ditujukan pada dua area yaitu kualitas produk dan kualitas proses. Kualitas perangkat lunak harus dikaji dengan benar sebagai persyaratan berdasarkan kehandalan, fungsionalitas, performa aplikasi, dan kinerja sistem.