

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Perangkat Lunak

Perangkat Lunak adalah perangkat yang ada di dalam komputer yang tidak dapat disentuh maupun lihat bentuk fisiknya. Namun, dapat dioperasikan pada saat menggunakan komputer atau media elektronik lainnya. Dalam pembuatan perangkat lunak memerlukan “bahasa pemrograman” yang ditulis oleh *programmer* kemudian dikompilasi dengan aplikasi *compiler* sehingga menjadi kode yang bisa dikenali oleh perangkat keras.

Pada awalnya perangkat lunak hanya menggunakan sistem benar atau salah atau sistem bilangan *binary* digit (bit) 1 dan 0. Pada awalnya bisa dilakukan tanpa kendala, namun lama kelamaan menjadi rumit untuk mengembangkan perangkat lunak hanya dengan sistem biner. Kemudian, pengelompokkan biner menjadi nibble (4 bit), byte (8 bit), word (16 byte), doubleword (32 bit). Pengelompokkan bit ini disusun ke dalam struktur instruksi seperti penyimpanan, transfer, operasi aritmatika, operasi logika, dan bentuk bit ini diubah menjadi kode-kode yang dikenal sebagai assembler. Pada akhirnya dikembangkan menjadi bahasa pemrograman tingkat tinggi yaitu bahasa manusia dalam bahasa Inggris [1].

Perkembangan perangkat lunak dibagi menjadi empat era, yaitu [1]:

1. Era Pioner

Pada era pioner perangkat lunak hanyalah sebuah sambungan-sambungan antar bagian dalam komputer. Penggunaan komputer saat itu masih dilakukan secara langsung. Pada era ini, perangkat lunak merupakan satu kesatuan dengan perangkat kerasnya. Penggunaan komputer dilakukan secara langsung dan hasil yang selesai dikerjakan komputer berupa *print out*. Proses yang dilakukan di dalam komputer berupa baris instruksi yang secara berurutan diproses.

2. Era Stabil

Pada era stabil penggunaan komputer sudah banyak digunakan, tidak hanya oleh kalangan peneliti dan akademi saja, tetapi juga oleh kalangan

industri/perusahaan. Perusahaan perangkat lunak bermunculan, dan sebuah perangkat lunak dapat



# UNIVERSITAS MIKROSKIL

menjalankan beberapa fungsi, kemudian perangkat lunak mulai bergeser menjadi sebuah produk. Baris-baris perintah perangkat lunak yang dijalankan oleh komputer bukan lagi satu-satu, tapi sudah seperti banyak proses yang dilakukan secara serentak (*multitasking*). Pada era ini mulai dikenal sistem basis data, yang memisahkan antara program (pemroses) dengan data (yang diproses).

### 3. Era Mikro

Sejalan dengan semakin luasnya PC dan jaringan komputer di era ini, perangkat lunak juga berkembang untuk memenuhi kebutuhan perorangan. Perangkat lunak dapat dibedakan menjadi perangkat lunak sistem yang bertugas menangani internal dan perangkat lunak aplikasi yang digunakan secara langsung oleh penggunanya untuk keperluan tertentu. Otomatisasi yang ada di dalam perangkat lunak mengarah ke suatu jenis kecerdasan buatan.

### 4. Era Modern

Saat ini perangkat lunak sudah terdapat di mana-mana, tidak hanya pada sebuah super komputer dengan 25 prosesor, sebuah komputer genggam pun telah dilengkapi dengan perangkat lunak yang dapat disinkronkan dengan PC. Tidak hanya komputer, bahkan peralatan seperti telepon, TV, hingga ke mesin cuci, AC dan microwave, telah ditanamkan perangkat lunak untuk mengatur operasi peralatan itu. Dan suatu saat setiap peralatan itu akan dapat saling terhubung. Pembuatan sebuah perangkat lunak bukan lagi pekerjaan sedikit orang, tetapi telah menjadi pekerjaan banyak orang dengan beberapa tahapan proses yang melibatkan berbagai disiplin ilmu dalam perancangannya. Tingkat kecerdasan yang ditunjukkan oleh perangkat lunak semakin meningkat, selain permasalahan teknis, perangkat lunak sekarang mulai bisa mengenal suara dan gambar.

Ada beberapa bagian dalam perangkat lunak, yaitu[1]:

#### 1. Sistem operasi

Program yang ditulis untuk mengendalikan dan mengkoordinasikan kegiatan dari sistem komputer.

#### 2. Bahasa pemrograman

Program yang digunakan untuk menerjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman kedalam bahasa mesin supaya dapat dimengerti oleh komputer.

### 3. Aplikasi

Program yang ditulis dan diterjemahkan oleh bahasa pemrograman untuk menyelesaikan aplikasi.

#### 2.2 Aplikasi Adobe *Experience Design CC*

Perusahaan perangkat lunak terbesar dunia yaitu *Adobe System* meluncurkan program untuk para desainer aplikasi *mobile* dalam mengembangkan UI dan UX. Program ini adalah *Adobe Experience Design* atau dikenal dengan Adobe XD. Tidak diragukan lagi semua paket *software* desain dari Adobe selalu memiliki keunggulan tersendiri. Adobe XD adalah perangkat lunak perancang desain untuk aplikasi *mobile* dan web yang berbasis vektor. Dengan keberadaan Adobe XD mempermudah para perancang desain aplikasi *mobile* dalam menggarap pekerjaannya. Perangkat lunak ini mendukung *wireframing* situs web dan pembuatan prototipenya. Adobe menyediakan penawaran gratis pada Adobe XD untuk membuat UI, UX, dan *prototyping* ke dalam berbagai *platform* meliputi web, ponsel, tablet, dan sejenisnya[2].

Pengguna dapat membuat desain *interface* aplikasi sederhana sebagai prototipe sebelum diimplementasikan. Adobe XD CC didesain dan dibuat oleh Adobe System sehingga mendukung format *file* yang dibuat oleh aplikasi Adobe lainnya, seperti Adobe Photoshop dan Adobe Illustrator. Adobe XD CC merupakan perangkat lunak yang masih tergolong baru. Sebelumnya aplikasi ini dikenal dengan nama Project Comet pada April 2016 yang kemudian berubah nama menjadi Adobe Experience Design dan memasuki tahap Beta sejak September 2016 yang kemudian resmi diluncurkan pada bulan Oktober 2017[3].

Terdapat beberapa fitur Adobe XD yang ditawarkan dalam membuat UI dan UX, antara lain[4]:

#### 1. Visualisasi *Flow* dan *Trigger*

Fitur Adobe XD tidak hanya dapat digunakan untuk mendesain, tetapi dapat juga digunakan untuk membuat prototipe. Adobe XD juga menawarkan visualisasi *flow* dan *trigger* secara langsung.

#### 2. Menggambar Vektor

Fitur Adobe XD mampu menggambar atau menciptakan sebuah bentuk untuk membuat *wireframe*, ikon, komponen, serta elemen-elemen visual lainnya.

### 3. Interaksi Mikro Otomatis

Interaksi mikro adalah salah satu fitur yang dapat meningkatkan user experience, sehingga dapat membantu menciptakan animasi-animasi secara otomatis. Fitur ini bisa membuat proses desain produk menjadi lebih lancar.

### 4. Terdapat *Asset Libraries*

Dalam pembuatan aplikasi membutuhkan tampilan visual dan warna-warna yang senada agar tampilan lebih menarik. Adobe XD memberikan fitur *asset libraries* yang memudahkan proses penyesuaian pembuatan desain visualisasi produk.

### 5. Integrasi dengan Aplikasi Lain

Adobe XD menawarkan banyak *plugin* yang bisa langsung dipasang. Adobe XD juga menawarkan fitur edit foto yang bisa disinkronisasi langsung menggunakan Adobe Photoshop. Bahkan, dapat mengimpor aneka desain dari Adobe Sketch, Illustrator, serta Photoshop itu sendiri.

### 6. Fitur Kolaborasi

Biasanya proses desain UX tidak hanya dilakukan oleh satu orang saja, melainkan harus kolaborasi dengan banyak orang. Adobe XD, memberikan fitur dalam melakukan editing satu dokumen dapat lebih dari satu orang secara bersama-sama, tentunya hal ini dapat memudahkan proses kolaborasi antar anggota tim.

## 2.3 Aplikasi *Mobile* Android

Aplikasi *mobile* adalah sebuah perangkat lunak yang dirancang yang dapat berjalan pada beberapa atau semua perangkat seluler seperti ponsel pintar dan tablet. Pernyataan ini berdasarkan hasil inovasi teknologi baru-baru ini. Aplikasi *mobile* muncul dikarenakan adanya konvergensi media, teknologi informasi, internet dan teknologi canggih lainnya[5].

Beberapa tahun terakhir, dunia dihebohkan dengan adanya *platform* baru yang semakin menguasai pasar global bahkan saat ini lebih dari setengah persen pengguna ponsel dunia telah menggunakan sistem operasi tersebut. Sistem operasi tersebut adalah Android. Meski sebagai *platform* pendatang baru, terhitung sejak

pertengahan tahun 2013 telah mencapai 79% *market share*. Hal tersebut tidak terlepas dari adanya kerjasama antara pihak Android dengan berbagai perusahaan teknologi raksasa, seperti Samsung, Asus, MITO, Cross, HTC, Sony, HTC, Oppo, Nokia, Coolpad, Lenovo, dan Motorola yang saat ini telah menggandeng Android sebagai sistem operasi ponsel mereka[6].

Android memiliki beberapa keunggulan bila dibandingkan dengan sistem operasi lainnya yaitu sebagai berikut[6].

a. Cepat dan Responsif

Android dapat dinobatkan sebagai sistem operasi *open source* yang cepat dan responsif pula. Studi *Blancco Technology Group* merilis bahwa *crash* yang terjadi, baik pada aplikasi, konektivitas *Wi-Fi*, dan masalah kinerja lainnya yang menyebabkan kegagalan pada sistem operasi Android lebih minim daripada sistem operasi lainnya yaitu hanya memiliki kegagalan rata-rata berada di angka 35%.

b. Mudah digunakan / *User Friendly*

Mulai dari tampilan sampai cara kerja sistem operasinya, Android tergolong sistem operasi yang mudah digunakan tanpa keahlian khusus untuk pengoperasian. Itulah alasan utama kenapa Android cepat diterima oleh semua lapisan generasi, mulai dari anak usia belia hingga lanjut usia.

c. Variasi harga produk yang beragam

Dikarenakan sistem operasi Android digunakan oleh berbagai produsen ponsel dunia membuat Android memiliki harga yang merakyat dan dapat dinikmati oleh semua golongan.

d. Google sebagai pengembang

Android berada pada naungan Google membuat Android juga mendapatkan dampak yang positif dari Google. Google begitu responsif dengan berbagai perkembangan teknologi yang ada sehingga hal tersebut memastikan bahwa sistem operasi Android akan terus menjadi sistem operasi paling subur dengan jumlah *device* akses yang mencapai 85% dengan publikasi yang sangat baik.

e. *Hardware* pendukung yang beragam

Meskipun lebih dikenal sebagai sistem operasi pada ponsel pintar, Android juga dikembangkan dan dapat diaplikasikan di berbagai peralatan perangkat keras lainnya, seperti jam tangan, tablet, PC, dan *Smart TV*.

Aplikasi *mobile* android dari segi *front-end* dapat dikembangkan dengan menggunakan perangkat lunak *React-Native*.

*React-Native* adalah *framework* atau kerangka kerja *JavaScript* yang digunakan untuk mengembangkan aplikasi *mobile* yang nyata dan asli untuk iOS dan Android. Kerangka kerja ini didasarkan pada *React* dengan menggunakan *library JavaScript* dari Facebook untuk membangun antarmuka pengguna. Mirip dengan *ReactJs*, *React-Native* juga ditulis menggunakan campuran *JavaScript* dan XML (*Extensible Markup Language*) yang dikenal sebagai *JSX (JavaScript XML)*. *React-Native* mengekspos antarmuka *JavaScript* untuk API platform sehingga aplikasi *React-Native* dapat mengakses fitur platform seperti kamera ponsel dan lokasi pengguna.

*React-Native* mendukung sistem operasi iOS dan Android yang memiliki potensi untuk berkembang ke platform masa depan juga. *React-Native* menerjemahkan *markup* menjadi elemen antarmuka pengguna asli yang nyata dengan memanfaatkan sarana *rendering* tampilan yang ada pada platform apapun yang digunakan. Selain itu, *React-Native* juga bekerja secara terpisah dari utas antarmuka pengguna sehingga aplikasi dapat mempertahankan kinerja yang tinggi tanpa mengorbankan kemampuan[7].

## 2.4 Situs Web

Situs web adalah dokumen yang tersimpan dalam *web server* yang memiliki jenis dan topik tertentu, pada umumnya situs web merupakan kumpulan halaman web yang saling terhubung dan berkas-berkasnya saling terkait. Jenis situs web saat ini sangat beragam dan disesuaikan dengan fungsinya masing-masing. Terdapat beberapa situs web yang sukses berkat besarnya jumlah komunitas penggunanya. Situs web yang meraih kesuksesan besar berkat komunitas penggunanya di antaranya adalah Kaskus, Facebook dan Twitter. Situs tersebut mampu mengambil kesempatan untuk memanfaatkan kecenderungan manusia untuk berinteraksi sosial dengan orang lain.

Kecenderungan manusia untuk saling berbagi juga mampu secara sempurna diakomodasi oleh keberadaan situs web tersebut. Hal tersebut menunjukkan bahwa manusia selaku makhluk sosial mempunyai dorongan untuk saling berbagi. Situs web dapat mempermudah para peneliti ataupun masyarakat luas untuk mendapatkan informasi berbentuk jurnal dalam satu wadah yang disediakan. Jadi ini lebih efektif dan efisien dalam hal pencarian dan waktu yang digunakan dalam proses pencarian sumber-sumber referensi yang dibutuhkan[8].

## 1. HTML

*HyperText Markup Language* (HTML) merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman web yang dapat diakses atau ditampilkan menggunakan *Web Browser*. HTML sendiri secara resmi dibuat pada tahun 1989 oleh Tim Berners Lee dan dikembangkan oleh *World Wide Web Consortium* (W3C), yang kemudian pada tahun 2004 dibentuklah *Web Hypertext Application Technology Working Group* (WHATWG) yang hingga kini bertanggung jawab akan perkembangan bahasa HTML ini. Hingga kini telah mengembangkan HTML 5, sebuah versi terbaru dari HTML yang mendukung tidak hanya gambar dan teks, namun juga menu interaktif, audio, video dan lain sebagainya.

Dalam ilmu pemrograman sering dikenal istilah *compile* agar dapat menjalankan hasil koding, namun dalam HTML tidak perlu memakai aplikasi untuk meng-*compile* file tersebut. Untuk menjalankan *file* HTML hanya perlu mengeksekusinya menggunakan aplikasi *Web Browser*[9].

## 2. CSS

*Cascading Style Sheet* (CSS) merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan halaman web supaya tampilan lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh W3C pada tahun 1996.

Pada awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa *markup* seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (*XML User Interface Language*). Pada bulan Desember 1996, W3C memperkenalkan Level 1 spesifikasi CSS atau juga dikenal CSS1 yang mendukung format, warna *font* teks,



dan lain-lain. Kemudian, bulan Mei 1998, W3C menerbitkan CSS2 yang di dalamnya diatur fungsi peletakan elemen. Dan sekarang, W3C telah memperbaiki dan meningkatkan Kemampuan CSS2 ke CSS3[9].

### 3. Bootstrap

Rancang bangun web *layout* merupakan salah satu faktor utama yang harus direncanakan dengan baik agar *website* dapat berjalan dengan baik, cepat dan dapat diakses oleh berbagai macam *browser* serta dapat berjalan diberbagai macam *platform*. Bootstrap merupakan sebuah *framework* yang dapat menyelesaikan permasalahan dalam mendesain web. Bootstrap juga responsif terhadap banyak *platform*, artinya tampilan halaman *website* yang menggunakan Bootstrap ini akan tampak tetap rapi, baik versi *mobile* maupun *desktop*[10].

Bootstrap adalah salah satu *framework* yang memungkinkan pengembang dapat mengembangkan situs web dengan mudah dan cepat. pengembang hanya perlu memanggil *class* tertentu untuk membuat tombol, panel, tabel, pesan peringatan, dan lain sebagainya. Bootstrap terdiri dari beberapa *file*. *File* pada Bootstrap berisi kumpulan baris kode tersusun dari CSS dan *JavaScript* yang berbentuk *class*. Bootstrap menyediakan fondasi yang kuat untuk situs web apa pun. Salah satu alasan menggunakan Bootstrap adalah karena menghemat pekerjaan[11].

Ada lima alasan Bootstrap lebih dimintai oleh para pengembang situs web, yaitu[11]:

#### a. Sistem grid yang *powerful*

Bootstrap memiliki fitur responsif terbaik. Dibuat dengan *flexbox* dan mudah digunakan. Hal ini sangat membantu dalam penskalaan satu desain situs web dari perangkat di berbagai ukuran. Membagi layar menjadi 12 kolom, sehingga dapat memutuskan berapa banyak kolom yang diambil untuk setiap elemen desain situs web.

#### b. Proses Pembuatan web yang cepat

Bootstrap dilengkapi dengan banyak CSS dan komponen *JavaScript* yang dapat digunakan untuk membantu mencapai fungsionalitas yang dibutuhkan di hampir semua jenis situs web. Kamu hanya perlu menggunakan beberapa HTML untuk memasukkannya ke dalam *script template*, tanpa perlu menghabiskan banyak

waktu menulis CSS dan JavaScript yang kompleks dan juga semua komponen ini juga responsif.

c. Kompatibilitas *Browser*

Bootstrap kompatibel dengan versi terbaru dari semua *browser*. Untuk Internet Explorer, Bootstrap bekerja di Internet Explorer versi 10 dan 11.

d. Mudah dikustom

Bootstrap menawarkan banyak cara untuk menyesuaikan desain utamanya. Bahkan untuk yang biasanya menggunakan SASS, Bootstrap menyertakan opsi penyesuaian SASS.

e. *Open Source*

Bootstrap adalah proyek *open source* yang dirilis di bawah lisensi MIT. Lisensi Bootstrap memberikan kebebasan untuk sepenuhnya mengubah dan bereksperimen dengannya.

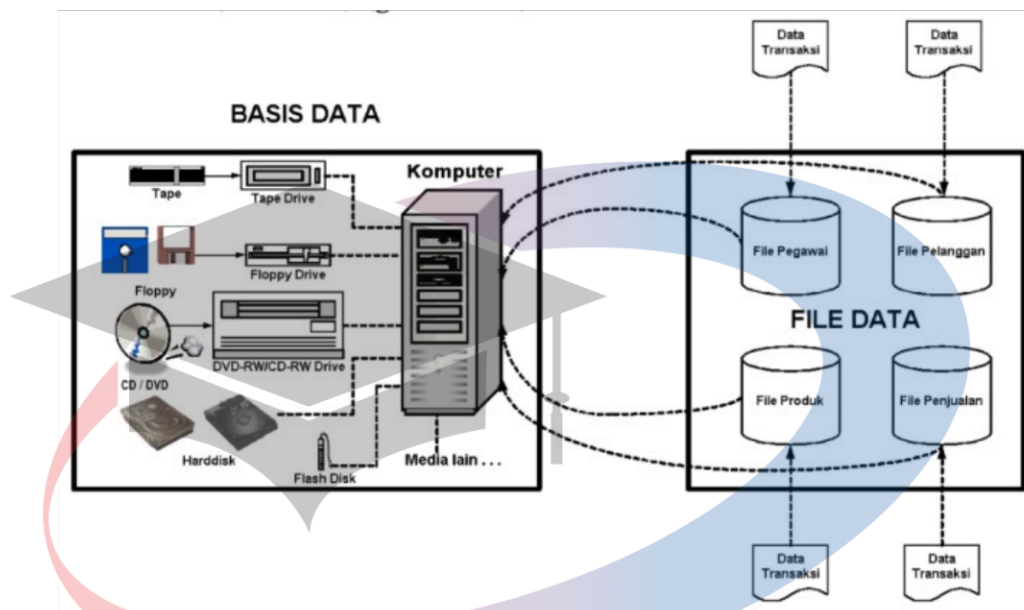
4. JavaScript

JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. JavaScript merupakan perekat yang menyatukan halaman-halaman web. JavaScript dapat digunakan untuk mengendalikan tampilan halaman. JavaScript dipakai untuk memvalidasi apa yang diketikkan pengguna ke dalam sebuah *form* sebelum pengiriman *form* ke *server* dilakukan. JavaScript berperan sebagai bahasa pemrograman, yang memiliki konstruksi-konstruksi dasar seperti variabel dan tipe data, *look* kontrol, *statement if* atau *else*, *statement switch*, fungsi, dan objek. JavaScript dapat dipakai untuk perhitungan aritmatik, manipulasi tanggal dan waktu, pemodifikasian array, string dan objek. Ia juga dapat menangani *event* yang diinisiasi pengguna, dan menetapkan pewaktu[12].

## 2.5 Basis Data

Data didapat dari suatu kejadian yang benar-benar terjadi misalnya data penjualan didapat dari data hasil penjualan. Data dalam suatu perusahaan dapat disamakan dengan laporan tertulis baik antar departemen maupun dari luar perusahaan. Setelah semua fakta telah diterima, fakta tersebut akan dimasukkan dan dikelompokkan ke dalam data. Kemudian data tersebut akan dikelompokkan dan

dimasukkan ke dalam *file* yang disebut dengan *file data*. Dari *file data* tersebut barulah dimasukkan dan dihubungkan antara *file data* yang satu dengan yang lain dalam sebuah tempat penampungan data yang dikenal dengan basis data[13].



Gambar 2.1 Basis Data

*Database* atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*Database Management System, DBMS*). Sistem basis data dipelajari dalam ilmu informasi.

Secara konsep basis data adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi. Pada komputer, basis data disimpan dalam perangkat *hardware* penyimpanan, dan dengan *software* tertentu dimanipulasi untuk kepentingan atau kegunaan tertentu. Hubungan atau relasi data biasanya ditunjukkan dengan kunci (*key*) dari setiap *file* yang ada. Data merupakan fakta atau nilai (*value*) yang tercatat atau merepresentasikan deskripsi dari suatu objek. Data yang merupakan fakta yang tercatat dan selanjutnya dilakukan pengolahan (proses)

menjadi bentuk yang berguna atau bermanfaat bagi pemakainya akan membentuk apa yang disebut informasi.

Suatu bangunan basis data biasanya memiliki 5 (lima) jenjang yaitu sebagai berikut[14]:

1. Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau karakter khusus yang membentuk sebuah item data atau *field*.
2. *Field* atau Item, merupakan representasi suatu atribut dan *record* yang sejenis yang menunjukkan suatu item dari data.
3. *Record* (rekaman atau tupel), Kumpulan dari *field* membentuk suatu *record* atau rekaman. *Record* menggambarkan suatu unit data individu yang tertentu.
4. *File*, merupakan kumpulan dari rekaman yang menggambarkan satu kesatuan data sejenis yang mewakili tiap-tiap data.
5. *Database*, merupakan kumpulan dari *file* atau tabel yang membentuk suatu *database*.

Dalam pembuatan dan penggunaan basis data, terdapat 4 (empat) komponen dasar sistem basis data, yaitu[13]:

#### 1. Data

Data yang digunakan dalam sebuah basis data, haruslah mempunyai ciri sebagai berikut:

- a. Data disimpan secara terintegrasi (*integrated*), yaitu basis data merupakan kumpulan dari berbagai macam file dari aplikasi-aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian-bagian yang rangkap (*redundant*).
- b. Data dapat dipakai secara bersama-sama (*Shared*), yaitu masing-masing bagian dari basis data dapat diakses oleh pemakai dalam waktu yang bersamaan, untuk aplikasi yang berbeda.

#### 2. Hardware

Terdiri dari semua peralatan perangkat keras komputer yang digunakan untuk pengelolaan sistem *database*, seperti:

- a. Peralatan untuk penyimpanan, disk, drum, dll.
- b. Peralatan *input* dan *output*.
- c. Peralatan komunikasi data, dll.

### 3. *Software*

Berfungsi sebagai perantara (*interface*) antara pemakai dengan data fisik pada database, dapat berupa:

- a. *Database Management System* (DBMS)
- b. Program-program aplikasi & prosedur-prosedur yang lain, seperti Oracle, SQL Server, MySQL, dll

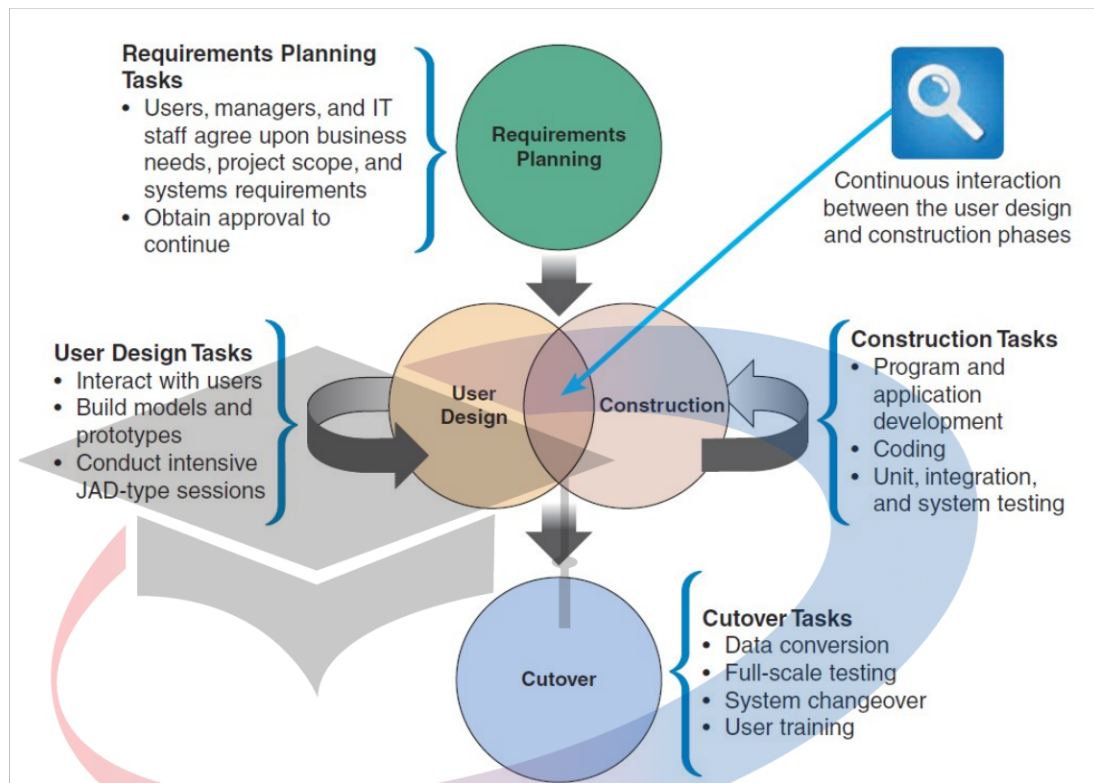
### 4. *User* (Pengguna)

Terbagi menjadi 3 klasifikasi :

- a. *Database Administrator* (DBA), yaitu orang atau tim yang bertugas mengelola sistem basis data secara keseluruhan.
- b. *Programmer*, yaitu orang atau tim yang membuat program aplikasi yang mengakses basis data dengan menggunakan bahasa pemrograman.
- c. *End user*, orang yang mengakses basis data melalui terminal dengan menggunakan *query language* atau program aplikasi yang dibuat oleh *programmer*.

## 2.6 Rapid Application Development

RAD adalah sebuah metodologi lengkap yang mempunyai 4 tahapan siklus hidup yang bersifat paralel RAD menggunakan teknik berbasis tim sehingga dapat mempercepat pengembangan sistem informasi dan menghasilkan sistem informasi yang berfungsi. Biasanya perusahaan menggunakan metode RAD untuk mengurangi biaya dan waktu dalam pengembangan serta meningkatkan kemungkinan berhasilnya pengembangan. RAD sangat bergantung terhadap pembuatan prototipe dan keterlibatan dari pengguna. Proses RAD memungkinkan pengguna untuk memeriksa model kerja sedini mungkin, menentukan apakah pengembangan tersebut dapat memenuhi kebutuhan mereka, dan menyarankan perubahan yang diperlukan. Berdasarkan masukan dari pengguna, prototipe akan di modifikasi. Proses interaktif tersebut akan berlanjut hingga sistem sepenuhnya dikembangkan dan pengguna puas terhadap hasil sistem yang dibuat. Tim proyek menggunakan *tools* untuk membangun prototipe dan membuat aliran dokumentasi yang berkelanjutan[15].



Gambar 2.2 Tahapan *Rapid Application Development*

RAD melibatkan pengguna di setiap bagian dari upaya pengembangan, dengan partisipasi yang intens di bagian bisnis dari desain. Ada empat tahapan untuk RAD yang melibatkan pengguna dan analisis yaitu sebagai berikut[15].

1. Tahap Perencanaan Kebutuhan (*Requirements Planning*)

Tahap perencanaan kebutuhan menggabungkan elemen-elemen dari perencanaan sistem dan tahap analisis sistem dari *Systems Development Life Cycle* (SDLC). Pengguna, manajer, dan anggota staf TI mendiskusikan dan menyepakati bersama kebutuhan dari bisnis, ruang lingkup, kendala, dan persyaratan sistem.

2. Tahap Desain Pengguna (*User Design*)

Pada tahap desain pengguna, analisis sistem berinteraksi dengan pengguna untuk mengembangkan model dan prototipe yang mewakili semua proses sistem, *input*, dan *output*. Desain pengguna adalah proses interaktif berkelanjutan yang memungkinkan pengguna untuk memahami, memodifikasi, dan akhirnya menyetujui model kerja sistem yang memenuhi kebutuhan mereka.

### 3. Konstruksi (*Construction*)

Tahap konstruksi berfokus pada program dan tugas pengembangan aplikasi yang serupa dengan SDLC. Tetapi pada RAD, pengguna akan terus berpartisipasi dan masih dapat menyarankan perubahan atau peningkatan saat layar atau laporan aktual dikembangkan.

### 4. Peralihan (*Cutover*)

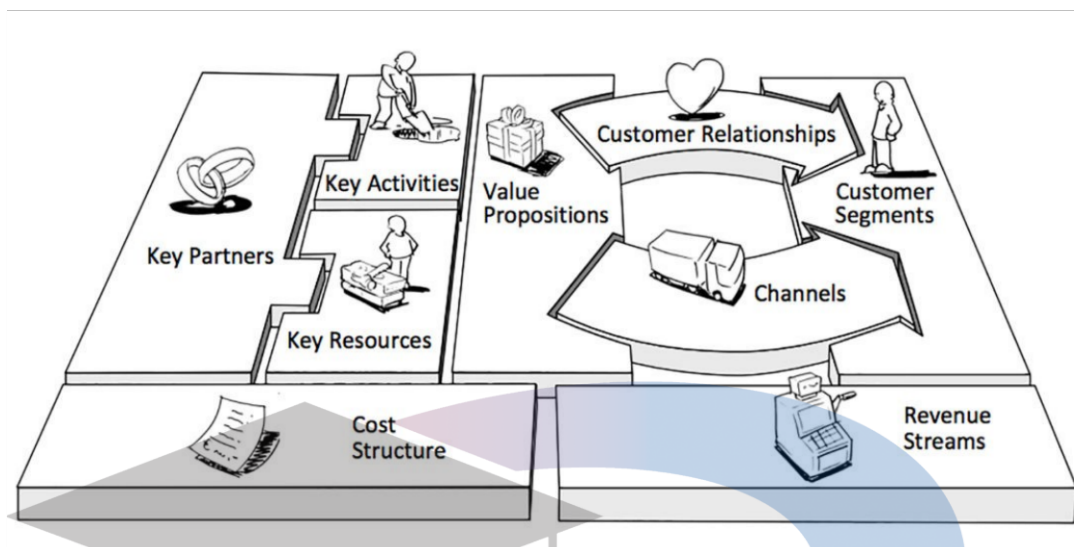
Tahap peralihan menyerupai fase dalam implementasi SDLC yang termasuk dalam konversi data dari sistem lama ke baru, pengujian, pergantian ke sistem baru, dan pelatihan pada pengguna. Dibandingkan dengan metode tradisional, seluruh proses akan dikompresi sehingga sistem yang baru dibangun, dikirim, dan dioperasikan lebih cepat.

Tujuan utama dari pendekatan RAD adalah untuk memangkas waktu dan biaya pengembangan dengan melibatkan pengguna dalam setiap tahap pengembangan sistem. Dikarenakan RAD merupakan proses yang berkelanjutan, RAD memungkinkan tim pengembangan untuk membuat modifikasi yang diperlukan dengan cepat seiring dengan berkembangnya desain. Penting juga untuk membatasi biaya perubahan pada saat anggaran perusahaan sedang ketat ketika perubahan terjadi dalam jadwal pengembangan yang panjang dan berlarut-larut. Selain keterlibatan pengguna, tim RAD yang sukses memiliki sumber daya, keterampilan, dan dukungan manajemen TI. Karena merupakan proses yang dinamis dan digerakkan oleh pengguna, RAD sangat berharga ketika perusahaan membutuhkan sistem informasi untuk mendukung fungsi bisnis baru. Dengan mendapatkan masukan pengguna dari awal, RAD juga membantu tim pengembangan merancang sistem yang membutuhkan antarmuka pengguna yang sangat interaktif atau kompleks[15].

## 2.7 Teknik Analisis dan Perancangan Aplikasi

### 2.7.1 Kanvas Model Bisnis

Kanvas Model Bisnis adalah salah satu alat untuk melihat lebih akurat bagaimana rupa bisnis yang sedang atau akan dijalani. Dengan alat ini kita seakan melihat bisnis dari gambaran besar namun tetap lengkap dan mendetail apa saja elemen penting yang terkait dengan bisnis[16].



Gambar 2.3 Kanvas Model Bisnis

BMC diciptakan oleh Alexander Osterwalder dalam bukunya yang berjudul *Business Model Generation*. BMC merupakan model bisnis yang dituangkan ke dalam visual gambar yang dibagi menjadi sembilan aspek bisnis yaitu[16]:

1. Segmen Pelanggan (*Customer Segments*)

Pelaku usaha harus menyadari bahwa pelanggan adalah jantung dari setiap bisnis. Tanpa adanya pelanggan, tidak ada satupun perusahaan yang dapat bertahan dalam jangka waktu yang lama. Dalam memenuhi kepuasan pelanggan, perusahaan mengelompokkan pelanggan menjadi beberapa segmen yang berbeda berdasarkan kesamaan kebutuhan, perilaku, dan lain sebagainya.

2. Nilai Unggul (*Value Proposition*)

Beberapa elemen yang berkontribusi pada pembentukan nilai unggul adalah :

- a. Kebaruan : proposisi nilai yang sebelumnya tidak pernah ditawarkan oleh perusahaan manapun, baik dari segi fitur, layanan baru dan sifat inovasinya. Hal ini biasanya terjadi pada produk teknologi.
- b. Kinerja : dapat meningkatkan kualitas produk ataupun kinerja dalam pelayanan merupakan cara yang umum untuk membuat nilai tambah.
- c. Kustomisasi : artinya produk atau jasa yang dapat disesuaikan dengan kebutuhan individual pelanggan. Saat ini banyak dikembangkan konsep *mass customization* yaitu bentuk penyesuaian kebutuhan individu tetapi tidak bisa diproduksi secara massal.



- d. Menyelesaikan Pekerjaan : nilai tambah ini dapat dibuat dengan cara membantu pelanggan dalam menyelesaikan beberapa pekerjaan.
- e. Desain : merupakan atribut yang penting namun sulit untuk diukur. Sebuah produk dapat tampak mewah karena desain yang bagus. Dalam industri fashion dan elektronik, desain adalah nilai unggul yang sangat penting.
- f. Merek atau Status: pelanggan dapat menemukan nilai tambah dalam merek dari suatu produk ataupun jasa.
- g. Harga : Perusahaan yang menawarkan produk yang mirip dengan produk yang telah ada dipasar dan lebih murah merupakan cara umum untuk memenangkan segmen pasar yang sensitif terhadap harga.
- h. Pengurangan Biaya : perusahaan memberikan nilai berupa pengurangan biaya dari aktivitas yang dilakukan oleh pelanggan.
- i. Pengurangan Risiko : dengan cara mengurangi risiko yang dihadapi pelanggan merupakan nilai tambah bagi perusahaan.
- j. Akses : untuk memberikan nilai tambah dengan membuat produk atau jasa yang dapat dijangkau oleh pelanggan dimana pelanggan sebelumnya tidak dapat menjangkau produk atau jasa tersebut.
- k. Kenyamanan/Kemudahan Penggunaan : Nilai tambah yang sangat penting bagi perusahaan adalah dengan menciptakan suatu produk atau jasa yang lebih nyaman dan mudah digunakan oleh pelanggan.

### 3. Saluran (*Channel*)

Dalam konsep BMC, saluran merupakan wadah untuk menyampaikan produk atau jasa dengan nilai unggul kepada pasar sasaran yang akan dituju. Agar dapat berfungsi secara optimal, saluran perlu didesain dengan mempertimbangkan aspek efektifitas dan efisiensi. Hal penting yang perlu diperhatikan pengusaha dalam membuat saluran harus mempertimbangkan kemampuan internal perusahaan dan juga perilaku konsumen dari pasar sasaran.

### 4. Hubungan Pelanggan (*Customer Relationship*)

Sebuah bisnis harus memiliki kejelasan tentang jenis hubungan yang ingin dibangun bersama kepada segmen pelanggan. Hubungan dapat berbentuk proposal sampai menggunakan sistem robot atau otomasi.

### 5. Sistem Pendapatan (*Revenue Stream*)

Sumber pendapatan berguna bagi pengusaha untuk merancang arus pendapatan yang akan diterima nantinya.

#### 6. Sumber Daya Utama (*Key Resources*)

Sumber daya utama yang memungkinkan organisasi menjalankan aktivitas utama yang akan membantu dalam menawarkan nilai unggul, menjangkau pasar, membangun hubungan dengan segmen pasar, dan memperoleh pendapatan. Sumber daya utama dapat berbentuk fisik (teknologi, mesin, atau peralatan, finansial, intelektual, manusia).

#### 7. Aktivitas Utama (*Key Activities*)

Aktivitas utama setiap bisnis berbeda-beda tergantung dengan bisnis apa yang dijalani. Beberapa aspek terkait aktivitas utama adalah kegiatan yang diperlukan untuk menghasilkan nilai unggul, menyalurkan nilai unggul ke pelanggan, kegiatan untuk menjalin hubungan dengan pelanggan, dan menangani aliran pendapatan. Aktivitas utama harus menunjukkan kegiatan operasional yang harus dilakukan dalam mewujudkan suatu produk atau jasa yang unggul sesuai dengan segmen pasar yang didasar.

#### 8. Mitra Utama (*Key Partners*)

Kemitraan dibangun untuk mendapatkan sumber daya yang tidak dimiliki. Untuk mencapai efisiensi terkadang harus menyerahkan kepada mitra untuk menangani salah satu aktivitas pada bisnis.

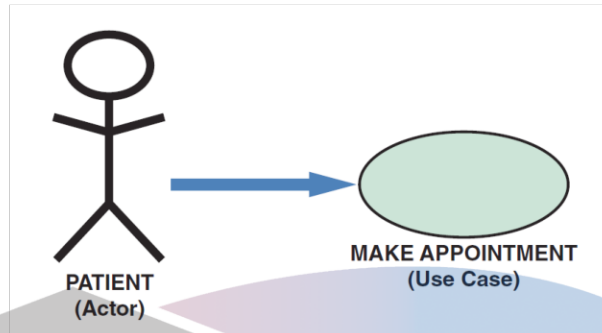
#### 9. Struktur Biaya (*Cost Structure*)

Struktur biaya yang berlaku pada umumnya dalam suatu sistem akuntansi adalah biaya tetap dan biaya variabel. Pada elemen ini menunjukkan biaya-biaya yang muncul ketika bisnis dijalankan. Dengan menciptakan dan memberikan nilai, mempertahankan hubungan pelanggan, dan menghasilkan pendapatan akan menyebabkan timbulnya biaya.

### 2.7.2 Pemodelan *Use Case* (*Use case Diagram*)

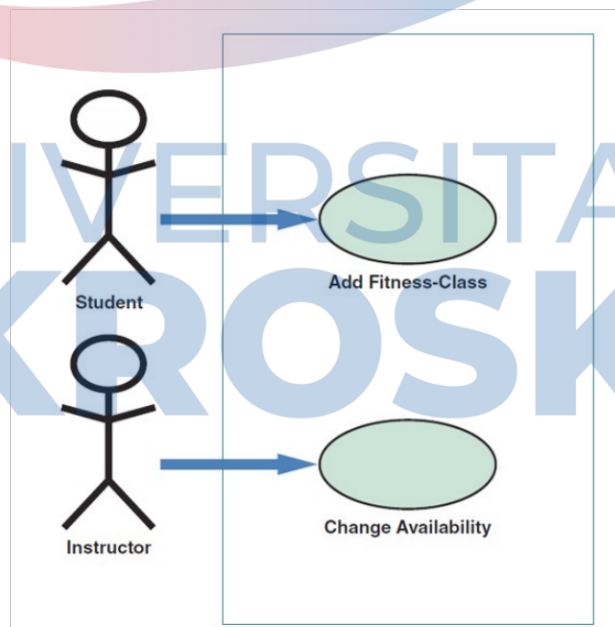
*Use case diagram* secara visual mewakili interaksi antara pengguna dan sistem informasi. Dalam *use case diagram*, pengguna menjadi Aktor, dengan peran tertentu yang menggambarkan bagaimana Aktor berinteraksi dengan sistem. Analisis

sistem dapat menggambar *use case diagram* secara bebas atau menggunakan alat *case* yang mengintegrasikan *use case* ke dalam desain sistem secara keseluruhan[15].



Gambar 2.4 Contoh Aktor Meminta Sistem Melakukan Fungsi

Entitas dalam *use case diagram* disebut dengan Aktor yang dimulai dengan meminta sistem untuk melakukan suatu fungsi atau proses. Misalnya, dalam sistem kantor medis, pasien sebagai Aktor dapat membuat janji temu (*use case*). Aktor ditampilkan sebagai figur tongkat, dengan label yang mengidentifikasi peran Aktor. Garis dari Aktor ke *use case* disebut asosiasi, karena menghubungkan Aktor tertentu dengan *use case*. *Use case* juga dapat berinteraksi dengan *use case* lainnya[15].



Gambar 2.5 Contoh *Use Case Diagram*

*Use case diagram* merupakan ringkasan visual dari beberapa *use case* terkait dalam suatu sistem atau subsistem. Ketika *use case diagram* dibuat, langkah pertama adalah mengidentifikasi batas sistem (*boundary of system*), yang diwakili oleh persegi panjang. Batas sistem menunjukkan apa yang termasuk dalam sistem (di

dalam persegi panjang) dan apa yang tidak termasuk dalam sistem (di luar persegi panjang). Setelah batas sistem diidentifikasi, use case ditempatkan pada diagram, Aktor ditambahkan, dan hubungan ditampilkan[15].

ADD NEW STUDENT Use Case	
<b>Name:</b>	Add New Student
<b>Actor:</b>	Student/Manager
<b>Description:</b>	Describes the process used to add a student to a fitness-class
<b>Successful completion:</b>	<ol style="list-style-type: none"> <li>1. Manager checks FITNESS-CLASS SCHEDULE object for availability</li> <li>2. Manager notifies student</li> <li>3. Fitness-class is open and student pays fee</li> <li>4. Manager registers student</li> </ol>
<b>Alternative:</b>	<ol style="list-style-type: none"> <li>1. Manager checks FITNESS-CLASS SCHEDULE object for availability</li> <li>2. Fitness-class is full</li> <li>3. Manager notifies student</li> </ol>
<b>Precondition:</b>	Student requests fitness-class
<b>Postcondition:</b>	Student is enrolled in fitness-class and fees have been paid
<b>Assumptions:</b>	None

Gambar 2.6 Contoh Deskripsi Use Case

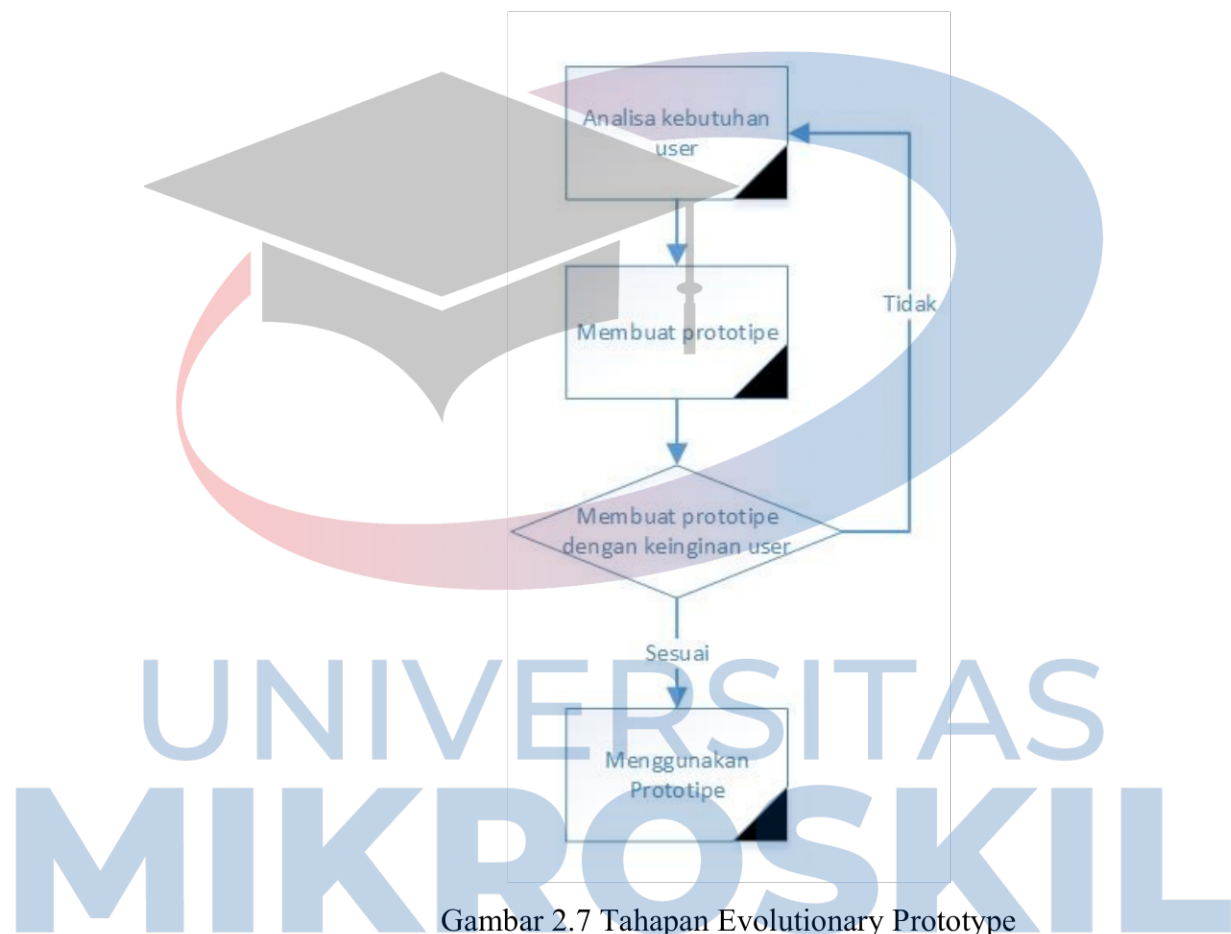
Untuk setiap *use case* dikembangkan deskripsi *use case* dalam bentuk tabel. Deskripsi *use case* mendokumentasikan nama *use case*, Aktor, deskripsi *use case*, daftar langkah demi langkah tugas dan tindakan yang diperlukan untuk penyelesaian yang berhasil, deskripsi tindakan alternatif, prakondisi, pascakondisi, dan asumsi[15].

### 2.7.3 Pembuatan Prototipe (*Prototyping*)

Pembuatan Prototipe merupakan teknik pengembangan sistem yang menggunakan prototipe untuk menggambarkan sistem, sehingga pengguna atau pemilik sistem mempunyai gambaran pengembangan sistem yang akan dilakukannya. Teknik ini sering digunakan apabila pemilik sistem tidak terlalu menguasai sistem yang akan dikembangkannya, sehingga dia memerlukan gambaran dari sistem yang akan dikembangkannya tersebut. Sehingga pengembang dapat membuat prototipe terlebih dahulu sebelum mengembangkan sistem yang

sebenarnya. Dalam pengembangan sistem, prototipe sering diwujudkan dalam bentuk *user interface* program aplikasi dan contoh-contoh laporan yang akan dihasilkan, sehingga dengan demikian pengguna sistem akan mempunyai gambaran tentang sistem yang akan digunakannya nanti[17].

Menurut McLeod dan Schell, tipe prototipe terbagi menjadi 2 (dua) yaitu sebagai berikut[17].



Gambar 2.7 Tahapan Evolutionary Prototype

1. *Evolutionary Prototype* adalah prototipe yang secara terus menerus dikembangkan hingga prototipe tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem.



Gambar 2.8 Tahapan *Requirement Prototype*

2. *Requirement Prototype* adalah prototipe yang dibuat oleh pengembang dengan mendefinisikan fungsi dan prosedur sistem dimana pengguna atau pemilik sistem tidak bisa mendefinisikan sistem tersebut.

Terdapat beberapa tahapan-tahapan dalam membuat prototipe yaitu sebagai berikut:

- a. Analisis kebutuhan *user*, pengembang dan pengguna atau pemilik sistem melakukan diskusi dimana pengguna atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.

- b. Membuat prototipe, pengembang membuat prototipe dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.
- c. Menyesuaikan prototipe dengan keinginan *user*, pengembang menanyakan kepada pengguna atau pemilik sistem tentang prototipe yang sudah dibuat, apakah sesuai dengan kebutuhan sistem atau tidak.
- d. Membuat sistem baru, pengembang menggunakan prototipe yang sudah dibuat untuk membuat sistem baru.
- e. Melakukan *testing* sistem, pengguna atau pemilik sistem melakukan uji coba terhadap sistem yang dikembangkan.
- f. Menyesuaikan dengan keinginan *user*, sistem disesuaikan dengan keinginan *user* dan kebutuhan sistem, jika sudah sesuai sistem siap digunakan.
- g. Menggunakan sistem.

Kelebihan dari penggunaan teknik pengembangan prototipe adalah menghemat waktu dan biaya pada saat pengembangan. Pengguna atau pemilik sistem bisa ikut terlibat dalam pengembangannya sehingga bisa meminimalisir terjadinya kesalahpahaman pada saat pengembangan sehingga menghasilkan kualitas sistem yang baik dan memungkinkan tim pengembang sistem untuk memprediksi dan memperkirakan pengembangan sistem selanjutnya.

Kelemahan dari penggunaan teknik pengembangan prototipe adalah, pengguna ataupun pemilik sistem bisa terus menerus menambahkan kompleksitas sistem sehingga sistem menjadi sangat kompleks, mengakibatkan sistem yang dikerjakan tidak akan pernah selesai[17].