

## BAB II LANDASAN TEORI

### 2.1 Metode *sequential*

*Sequential* adalah Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut: data yang ada dibandingkan satu persatu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan.

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke- $i$  dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk  $N$  elemen data harus dilakukan pencarian sebanyak  $N$  kali pula.

Data yang dicari di baca sebagai  $i$ ;

1.  $i = 0$  ←
2. Ketemu False. ←
3. Jika (Data [ $i$ ] = Key) Maka ketemu ← True jika tidak  $i$  ←  $i+1$ .
4. Jika (ketemu) maka  $i$  adalah indeks dari data yang dicari jika tidak data tidak ditemukan.

### 2.2 Kamus

(Hasan Alwi, 2003) mendefenisikan, Kamus merupakan khazanah perbendaharaan kata suatu bahasa yang menggambarkan tingkat peradaban bangsa pemiliknya.

(W.J.S Poerwadarminta, 2005) Kamus adalah buku yang berisi keterangan tentang arti kata-kata.

(Hoetomo M.A, 2005) mendefinisikan, Kamus yaitu buku acuan yang memuat kata dan ungkapan yang biasanya disusun menurut abjad berikut keterangan tentang maknanya.

Dari definisi kamus diatas dapat disimpulkan bahwa kamus merupakan sebuah buku yang memuat kata atau istilah yang disusun menurut abjad berikut keterangan dan maknanya

### **2.3 Desain Grafis**

Desain Grafis itu berasal dari dua kata yaitu Desain dan Grafis, kata Desain berarti proses atau perbuatan dengan mengatur segala sesuatu sebelum bertindak atau merancang. Sedangkan Grafis adalah titik atau garis yang berhubungan dengan cetak mencetak. Jadi dengan demikian Desain grafis ialah kombinasi kompleks antara kata-kata, gambar, angka, grafik, foto dan ilustrasi yang membutuhkan pemikiran khusus dari seorang individu yang bias menggabungkan elemen-elemen ini, sehingga mereka dapat menghasilkan sesuatu yang khusus atau sangat berguna dalam bidang gambar. Maria Regina College. 2008.

Dan Desain grafis merupakan suatu bentuk komunikasi visual yang menggunakan gambar untuk menyampaikan informasi atau pesan seefektif mungkin. Didalam desain grafis, teks juga dianggap gambar karena merupakan hasil abstraksi simbol-simbol yang bisa dibunyikan. Desain grafis pada awalnya diterapkan untuk media-media statis, seperti buku, majalah, dan brosur. Sebagai tambahan, sejalan dengan perkembangan zaman, desain grafis juga diterapkan dalam media elektronik yang sering disebut sebagai desain interaktif atau desain multimedia.

#### **2.3.1 Sejarah Desain Grafis**

Desain grafis berkembang pesat saat peradaban manusia menemukan tulisan dan mesin cetak. Berikut adalah peristiwa yang mempengaruhi perkembangan desain grafis di dunia:

##### **2.3.1.1 Tahun 1851 *The Great Exhibition***

Diselenggarakan di taman Hyde London antara bulan Mei hingga Oktober 1851, pada saat Revolusi industri. Pameran besar ini menonjolkan budaya dan industri serta merayakan teknologi industri dan disain. Pameran digelar dalam bangunan berupa struktur besi-tuang dan kaca, sering disebut juga dengan Istana Kristal yang dirancang oleh Joseph Paxton

#### **2.3.1.2 Tahun 1892, *Aristide Bruant, Henri Toulouse-Lautrec***

Pelukis *post-Impressionist* dan ilustrator art nouveau Prancis, Toulouse-Lautrec melukiskan banyak sisi Paris pada abad ke sembilan belas dalam poster dan lukisan yang menyatakan sebuah simpati terhadap ras manusia. Walaupun lithography ditemukan di Austria oleh Alois Senefelder pada tahun 1796, Toulouse-Lautrec membantu tercapainya peleburan industri dan seni.

#### **2.3.1.3 Tahun 1910, *Modernisme***

*Modernisme* terbentuk oleh urbanisasi dan industrialisasi dari masyarakat Barat. Sebuah *dogma* yang menjadi nafas desain modern adalah "Form *follow* Function" yang di lontarkan oleh Louis Sullivan. Simbol terkuat dari kejayaan *modernisme* adalah mesin yang juga diartikan sebagai masa depan bagi para pengikutnya. Desain tanpa dekorasi lebih cocok dengan "bahasa mesin", sehingga karya-karya tradisi yang bersifat *ornamental* dan *dekoratif* dianggap tidak sesuai dengan "estetika mesin"

#### **2.3.1.4 Tahun 1916, *Dadaisme***

Suatu pergerakan seni dan kesusasteraan (1916-1923) yang dikembangkan mengikuti masa Perang Dunia Pertama dan mencari untuk menemukan suatu kenyataan asli hingga penghapusan kultur tradisional dan bentuk estetik. *Dadaisme* membawa gagasan baru, arah dan bahan, tetapi dengan sedikit keseragaman. Prinsipnya adalah ketidakrasionalan yang disengaja, sifat yang sinis dan anarki, dan penolakan terhadap hukum keindahan.

#### **2.3.1.5 Tahun 1916, *De Stijl***

Gaya yang berasal dari Belanda, *De Stijl* adalah suatu seni dan pergerakan disain yang dikembangkan sebuah majalah dari nama yang sama ditemukan

oleh Theo Van Doesburg. De Stijl menggunakan bentuk segi-empat kuat, menggunakan warna-warna dasar dan menggunakan komposisi asimetris.

#### **2.3.1.6 Tahun 1918, *Constructivism***

Suatu pergerakan seni modern yang dimulai di Moscow pada tahun 1920, yang ditandai oleh penggunaan metoda industri untuk menciptakan object geometris. Constructivism Rusia berpengaruh pada pandangan *moderen* melalui penggunaan huruf *sans-serif* berwarna merah dan hitam diatur dalam blok *asimetris*.

#### **2.3.1.7 Tahun 1919, *Bauhaus***

*Bauhaus* dibuka pada tahun 1919 di bawah arahan arsitek terkenal Walter Gropius. Sampai akhirnya harus ditutup pada tahun 1933, *Bauhaus* memulai suatu pendekatan segar untuk mendisain mengikuti Perang Duni Pertama, dengan suatu gaya yang dipusatkan pada fungsi bukannya hiasan.

#### **2.3.1.8 Tahun 1928-1930, *Gill Sans***

*Tipograpfer Eric Gill* belajar pada Edward Johnston dan memperhalus tipe huruf *Underground* ke dalam *Gill Sans*. *Gill Sans* adalah sebuah jenis huruf *sans serif* dengan proporsi klasik dan karakteristik geometris lemah gemulai yang memberinya suatu kemampuan beraneka ragam (*great versatility*).

#### **2.3.1.9 Tahun 1931, *Harry Beck***

Perancang grafis Harry Beck (1903-1974) menciptakan peta bawah tanah London (London Underground Map) pada tahun 1931. Sebuah pekerjaan abstrak yang mengandung sedikit hubungan ke skala fisik. *Beck* memusatkan pada kebutuhan pengguna dari bagaimana cara sampai dari satu stasiun ke stasiun yang lain dan di mana harus berganti kereta.

#### **2.3.1.10 Tahun 1950, *International Style***

*International* atau *Swiss style* didasarkan pada prinsip *revolutioner* tahun 1920an seperti De Stijl, *Bauhaus* dan *Neue Typography*, dan itu menjadi resmi pada tahun 1950an. *Grid*, prinsip matematika, sedikit dekorasi dan jenis

huruf *sans serif* menjadi aturan sebagaimana *tipografi* ditingkatkan untuk lebih menunjukkan fungsi universal daripada ungkapan pribadi

#### **2.3.1.11 Tahun 1951, *Helvetica***

Diciptakan oleh Max Miedinger seorang perancang dari Swiss, *Helvetica* adalah salah satu tipe huruf yang paling populer dan terkenal di dunia. Berpenampilan bersih, tanpa garis-garis tidak masuk akal berdasarkan pada huruf *Akzidenz-Grotesk*. Pada awalnya disebut *Hass Grotesk*, nama tersebut diubah menjadi *Helvetica* pada tahun 1960. *Helvetica* keluarga mempunyai 34 model ketebalan dan *Neue Helvetica* mempunyai 51 model.

#### **2.3.1.12 Tahun 1960, *Psychedelia and Pop Art***

Kultur yang populer pada tahun 1960an seperti musik, seni, desain dan literatur menjadi lebih mudah diakses dan merefleksikan kehidupan sehari-hari. Dengan sengaja dan jelas, *Pop Art* berkembang sebagai sebuah reaksi perlawanan terhadap seni abstrak. Gambar dibawah adalah sebuah poster karya Milton Glaser yang menonjolkan gaya *siluet Marcel Duchamp* dikombinasikan dengan kaligrafi melingkar. Di cetak lebih dari 6 juta eksemplar

#### **2.3.1.13 Tahun 1984, *Émigré***

Majalah desain grafis Amerika, *Émigré* adalah publikasi pertama untuk menggunakan komputer *Macintosh*, dan mempengaruhi perancang grafis untuk beralih ke desktop publishing (DTP). Majalah ini juga bertindak sebagai suatu forum untuk eksperimen tipografi

## **2.4 Peringklanan**

Iklan merupakan salah satu media komunikasi pemasaran yang kerap digunakan dalam aktivitas ekonomi dalam upaya mengenalkan produk kepada konsumen. Situasi pasar yang kompetitif menjadikan tidak ada suatu bisnis yang mampu bertahan lama tanpa didukung oleh bauran komunikasi pemasaran semisal iklan yang efektif dan efisien. Iklan sebagai salah satu sarana pemasaran produk barang

atau jasa harus mampu tampil menarik dan persuasive. Maria Regina College. 2008.

## 2.5 Android

Safaat (2012) Mendefinisikan Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar (*smartphone*) dan komputer tablet. Sistem operasi ini identik dengan lambang robot hijau. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Karena merupakan *software* berbasis *open source* maka bagi seorang pengembang (*developer*) dapat memanfaatkannya secara gratis dengan melakukan perubahan dan pengembangan terhadap aplikasi yang akan digunakan. Sedangkan bagi produsen ponsel, keuntungan yang di peroleh yaitu berupa *efisiensi* karena produsen dapat memangkas biaya produksi.

### 2.5.1 Versi dan Jenis-Jenis Android

Pengembangan Android dimulai dengan berdirinya Android, Inc. Pada oktober 2003 dengan tujuan membuat *mobile device* yang lebih *smart* untuk menyaingi *Symbian* dan *Windows Mobile* yang populer saat itu (iPone dan Blackberry belum diliris). Pada tahun 2005, Android Inc. Diakuisisi oleh google. Pengembangan terus dilanjutkan sampai android versi beta diluncurkan pada tanggal 5 November 2007 bersamaan dengan berdirinya OHA (Open Handset Alliance). Dijelaskan pada tabel 2.1 pada halaman berikut:

**Tabel 2.1** Versi dan jenis-jenis *Android*

Versi	Nama	Rilis	Catatan
1.0	-	23 September 2008	Android pertama, hanya untuk <i>smartphone</i>
1.1	-	9 Februari 2009	
1.5	Cupcake	30 April 2009	Mulai pakai kode nama
1.6	Donut	15 September 2009	
2.0-2.1	Eclair	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	

2.2	(Frozen Yogurt)	20 Mei 2010	
2.3	Gingerbread	6 Desember 2010	Digunakan <i>smartphone</i> jenis lama
3.0-3.2	Honeycomb	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	ICS (Ice Cream Sandwich)	19 Oktober 2011	<i>Smartphone</i> dan Tablet
4.1-4.3	Jelly Bean	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	Update untuk memperbaiki dan menambah fitur-fitur ICS

**Tabel 2.1** Versi dan jenis-jenis *Android* (lanjutan)

Versi	Nama	Rilis	Catatan
4.4	Kit Kat	Oktober 2013	Diumumkan 3 September 2013
5.0	Lollipop	2014	Peningkatan system keamanan dan tampilan
6.0	Marshmallow	2ei 2015	

### 2.5.2 Kelebihan *Android*

*Android* memiliki kelebihan pada beberapa fitur diantaranya adalah:

- a. *User Friendly*, kalimat ini sangat melekat pada sistem operasi Windows milik microsoft, ibaratnya kita dengan sangat mudah mengoprasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada *Android* yang berjalan pada *Smartphone*.
- b. *Notifications*, kita dengan sangat mudah mendapatkan notifikasi dari *smartphone android* dengan mengatur beberapa akun *Email*, *SMS*, *Voice Dial*, *Update* dan lain sebagainya.
- c. Tampilan, dari segi tampilan *Android* tidak kalah bagusya dari *iOs* milik *Apple*, karena memang dari awal *android* hampir mengusung teknologi *iOs* hanya saja ini versi murahya.
- d. *Open Source*, *Operating system* ini memang dibuat *open source* oleh penciptanya, karena memang berbasis *kernel Linux*. Jadi jangan kaget

jika diluar sana ada banyak sekali *Custom Rom* untuk masing-masing perangkat *android*.

- e. Aplikasi, Untuk Aplikasi akan disajikan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar, dan kita bisa mendownloadnya di Google Play.

### **2.5.3 kekurangan *Android***

*Android* memiliki kekurangan pada beberapa fitur diantaranya adalah:

- a. *Update System*, untuk melakukan *update system* tidaklah mudah kita harus menunggu dari masing-masing *vendor* untuk merilis *Update Versi* yang terbaru.
- b. Baterai cepat habis, ini sering terjadi jika kita menyalakan paket data dan menggunakan *widget* serta aplikasi yang berjalan secara berlebihan, untuk mengatasinya kita harus mengurangi aktivitas aplikasi pada *smartphone*.
- c. Lemot atau Lag, ini ada kaitannya dengan spesifikasi dari masing-masing perangkat, namun ada kalanya *Android* tidak bersahabat dengan beberapa *aplikasi* dikarenakan Ram ataupun prosesornya yang kurang memadai, jadi sesuaikan aplikasi dengan perangkatnya.

## **2.6 *Adobe Photoshop***

Henky Prihatna (2005) Mendefinisikan *Photoshop* adalah salah satu *software* pengolah *grafik* yang banyak digunakan oleh para *desainer grafis* dan *web* di seluruh dunia. Tampilannya yang mudah dipahami, kelengkapan fasilitas yang ditawarkan, serta kemudahan memperoleh fasilitas pendukung dari berbagai sumber menjadikan *Photoshop* menjadi pilihan paling handal bagi para *desainer*.

## **2.7 *Eclipse***

*Eclipse* adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari *Eclipse*:

- a. *Multi-platform*: Target sistem operasi *Eclipse* adalah Microsoft



Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

- b. *Multi-language*: *Eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Perl, PHP, dan lain sebagainya.
- c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

*Eclipse* pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

## 2.8 SQLite

*SQLite* merupakan sebuah sistem manajemen basis data relasional berbasis ACID - *compliant* dan memiliki ukuran library yang relatif kecil. *Software* ini ditulis menggunakan bahasa C.

*SQLite* merupakan proyek yang bersifat publik domain yang dikerjakan oleh D, Richard Hipp. tidak seperti paradigma *client-server* umumnya, inti *SQLite* bukanlah sebuah sistem mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan.

Protokol komunikasi utama yang digunakan adalah melalui pemanggilan API langsung melalui bahasa pemrograman. mekanisme seperti ini tentu membawa keuntungan karena dapat mereduksi *overhead* dan *latency time*; intinya lebih sederhana.

Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan pada sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara

mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai. Salah satu keunggulan adalah hanya memakan sedikit memori saat dijalankan, hanya sekitar 250Kb sehingga cocok untuk piranti mobile. (Wahana, 2013). *SQLite* mendukung tipe data *text* (yang mirip dengan String di Java), *integer* (mirip dengan long di Java), serta *real* (mirip dengan double di Java).

## 2.9 UML (*Unified Modeling Language*)

Sulistyorini. P (2009) Mendefinisikan Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET.

### 2.9.1 Diagram UML

Model-model diagram dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram tersebut antara lain:

a. Diagram Kelas

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi.


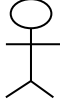

b. Diagram Paket

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

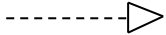
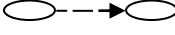
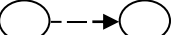
c. Diagram *Use case*

Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. *Elemen use case* terdiri dari:

**Tabel 2.2** Elemen *Use Case*

Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.




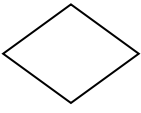


**Tabel 2.2** Elemen *Use Case* (Lanjutan)

Generalisasi 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.

- d. Diagram Interaksi atau *Sequence*  
Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
- e. Diagram Komunikasi  
Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirimi pesan.
- f. Diagram *Statechart*  
Bersifat dinamis. Diagram ini memperlihatkan keadaan-keadaan setiap sistem, memuat status, transisi, kejadian serta aktivitas.
- g. Diagram Aktivitas

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Pada tabel 2.2 berikut merupakan simbol – simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol – Simbol Aktivitas Diagram

Simbol	Keterangan
<i>Start state</i> 	Menunjukkan dimulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> .
<i>End state</i> 	Menggambarkan akhir atau terminal dari pada sebuah <i>activity diagram</i> .
<i>State transition</i> 	Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya
<i>Decision</i> 	Suatu titik/point pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.
<i>Activity</i> 	Menunjukkan aktivitas yang ada dalam system
<i>Swimlane</i> 	Membantu untuk lebih memahami siapa atau apa yang memulai aktivitas

#### h. Diagram Komponen

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem pada komponen-komponen yang telah ada sebelumnya

#### i. Diagram *Deployment*

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya.

### 2.9.2 Keunggulan UML

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur:

1. *Uniformity*

Pengembangan cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara terintegrasi bersama perancangan perangkat lunak dan perancangan struktur data.

2. *Understandability*

Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah untuk dipahami.

3. *Stability*

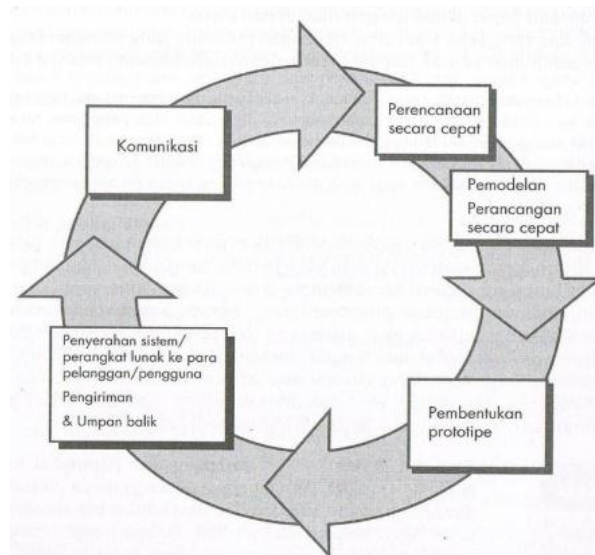
Kode program yang dihasilkan relatif stabil sepanjang waktu, karena mendekati permasalahan yang sesungguhnya.

4. *Reusability*

Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada akhirnya akan sangat mempercepat waktu pengembangan perangkat lunak (atau sistem informasi)

### 2.10 Metode *Prototype*

Pressman (2010), seringkali pelanggan mendefinisikan satu set tujuan umum untuk perangkat lunak, tetapi tidak mengidentifikasi persyaratan rinci untuk fungsi dan fitur. Di lain kasus, pengembang mungkin tidak yakin dari efisiensi dari sebuah algoritma, adaptasi dari sistem operasi, atau bentuk yang interaksi manusia-mesin harus ambil. Dalam hal ini, dan situasi lain, paradigma prototipe mungkin menawarkan pendekatan yang terbaik, *prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. pengembangan perangkat lunak. Seperti gambar 2.4 halaman berikut:



**Gambar 2.1** *Prototype* paradigma

*Prototype model* juga dapat didefinisikan sebagai proses pengembangan suatu *prototype* secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan sistem yang utuh. *Prototype model* merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototype* dapat dijelaskan sebagai berikut:

1. Pengumpulan kebutuhan
 

Pelanggan dan developer bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun prototyping
 

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).
3. Evaluasi prototyping
 

Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulangi langkah 1, 2, dan 3.
4. Mengkodekan sistem
 

Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

6. Evaluasi sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan, jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

### **2.10.1 Kelebihan dan Kekurangan Metode *Prototype***

Kelebihan *prototype* adalah:

- a. Adanya komunikasi yang baik antara pengembang dan pelanggan.
- b. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- c. Pelanggan berperan aktif dalam pengembangan sistem.
- d. Lebih menghemat waktu dalam pengembangan sistem.
- e. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Sedangkan kekurangan *prototype* adalah:

- a. Pelanggan tidak melihat bahwa perangkat lunak belum mencerminkan kualitas perangkat lunak secara keseluruhan dan belum memikirkan pemeliharaan dalam jangka waktu yang lama.
- b. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman sederhana.
- c. Hubungan pelanggan dengan komputer mungkin tidak menggambarkan teknik perancangan yang baik.



