

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Data Mining**

Menurut Turban et al mengatakan bahwa “Data mining merupakan suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan terkait dari berbagai database besar. Data mining juga diartikan sebagai suatu proses otomatis data-data yang sangat besar dan bertujuan untuk mendapatkan hubungan atau pola yang memberikan manfaat. Menurut Kusumo et al menjelaskan bahwa data mining juga merupakan suatu proses pendukung pengambil keputusan dimana mencari pola informasi dalam data. Pencarian ini dapat di lakukan oleh pengguna. Pencarian ini disebut discovery. Discovery merupakan proses pencarian dalam basis data dalam menemukan pola yang tersembunyi tanpa ide yang di dapatkan sebelumnya atau hipotesa tentang pola yang ada. Aplikasi mengambil inisiatif untuk menemukan pola dalam data tanpa pengguna berpikir mengenai pertanyaan yang relevan terlebih dahulu (*Sari,2018*).

Biasanya pengambilan keputusan ini berusaha untuk memanfaatkan gudang data yang sudah dimiliki untuk menggali informasi yang berguna membantu mengambil kesimpulan. Berdasarkan penjelasan mengenai pengertian data mining, dapat disimpulkan bahwa data mining merupakan suatu proses pengambilan suatu keputusan yang bertujuan untuk membentuk suatu hubungan yang memberikan manfaat didalam proses database.

#### **2.2 Pengelompokan Data Mining**

Menurut Larose data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan yaitu sebagai berikut:

## 1. Deskripsi

Terkadang peneliti dan Analisa secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

## 2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, peninjauan estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

## 3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Contohnya adalah pada prediksi harga beras dalam tiga bulan yang akan datang.

## 4. Klasifikasi

Dalam klasifikasi terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

## 5. Pengklusteran

Pengklusteran merupakan pengelompokkan record, pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.

## 6. Asosiasi

Tugas asosiasi dalam data mining adalah menemukan atribut yang muncul dalam satu waktu. Contohnya adalah pada meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler.

### 2.3 Clustering

Pada dasarnya clustering merupakan suatu metode untuk mencari dan mengelompokkan data yang memiliki kemiripan karakteristik (*similarity*) antara satu data dengan data yang lain. *Clustering* merupakan salah satu metode *data*

*mining* yang bersifat tanpa arahan (*unsupervised*), maksudnya metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*) serta tidak memerlukan target *output*. Dalam *data mining* ada dua jenis metode *clustering* yang digunakan dalam pengelompokan data, yaitu *hierarchical clustering* dan *non-hierarchical clustering*.

*Hierarchical clustering* adalah suatu metode pengelompokan data yang dimulai dengan mengelompokkan dua atau lebih objek yang memiliki kesamaan paling dekat. Kemudian proses diteruskan ke objek lain yang memiliki kedekatan kedua. Demikian seterusnya sehingga cluster akan membentuk semacam pohon dimana ada hierarki (tingkatan) yang jelas antar objek, dari yang paling mirip sampai yang paling tidak mirip. Secara logika semua objek pada akhirnya hanya akan membentuk sebuah cluster. Dendogram biasanya digunakan untuk membantu memperjelas proses hierarki tersebut.

Berbeda dengan metode *hierarchical clustering*, metode *non-hierarchical clustering* justru dimulai dengan menentukan terlebih dahulu jumlah *cluster* yang diinginkan (dua *cluster*, tiga *cluster*, atau lain sebagainya). Setelah jumlah *cluster* diketahui, baru proses *cluster* dilakukan tanpa mengikuti proses hierarki. Adapun metode ini biasa disebut dengan *K-Means Clustering*.

#### **2.4 Algoritma *K-Means Clustering***

*K-Means clustering* merupakan salah satu metode data *clustering* non-hirarki yang mengelompokkan data dalam bentuk satu atau lebih *cluster* atau kelompok. Data-data yang memiliki karakteristik yang sama dikelompokkan dalam satu *cluster* atau kelompok dan data yang memiliki karakteristik yang berbeda dikelompokkan dengan *cluster* yang lain sehingga data yang berada dalam satu *cluster* memiliki tingkat variasi yang kecil. Menurut Santosa (2007), langkah-langkah melakukan *clustering* dengan metode *K-Means* adalah sebagai berikut:

1. Pilih jumlah *cluster*.
2. Inisialisasi  $k$  pusat *cluster* ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara random. Pusat-pusat *cluster*

diberi nilai awal dengan angka-angka random.

3. Alokasikan semua data objek ke *cluster* terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data ke *cluster* tertentu ditentukan jarak antara data dengan pusat *cluster*. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat *cluster*. Jarak paling antara satu data dengan satu *cluster* tertentu menentukan suatu data masuk dalam *cluster* mana. Untuk menghitung jarak semua data ke setiap titik pusat *cluster* dapat menggunakan teori jarak *Euclidean* yang dirumuskan sebagai berikut:

$$D(i,j) = (x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2 \dots \text{dst.}$$

Keterangan:

$D(i,j)$  = Jarak antara data dengan titik pusat awal *cluster* tinggi

$i$  = Data Penjualan

$j$  = Titik pusat awal (*centroid*) *cluster* tinggi

$X_{ki}$  = Data ke  $i$  pada atribut data ke  $k$

$X_{kj}$  = Titik pusat ke  $j$  pada atribut ke  $k$

4. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/objek dalam *cluster* tertentu. Jika dikehendaki bisa juga menggunakan median dari *cluster* tersebut. Jadi rata-rata (*mean*) bukan satu-satunya ukuran yang bisa dipakai.
5. Tugaskan lagi setiap objek memakai pusat *cluster* yang baru. Jika pusat *cluster* tidak berubah lagi maka proses *clustering* selesai. Kembali ke langkah nomor 3 sampai pusat *cluster* tidak berubah lagi.

## 2.5 WEB Application

Menurut (Rizkita et al., 2018) web application atau aplikasi web wadah sebuah pekerjaan dengan menggunakan internet.

## 2.6 WEB

Menurut Sidik dalam Arizona (2017) mengatakan bahwa, "Situs Web (Website) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep

hiperlink yang memudahkan surfer ( sebutan bagi pemakai komputer yang melakukan penyelusuran informasi di Internet) untuk mendapatkan informasi dengan cukup mengklik suatu link berupa teks atau gambar maka informasi dari teks atau gambar akan ditampilkan secara lebih terperinci (detail)”.

## **2.7. Bahasa Pemrograman**

### **2.7.1 PHP**

Menurut Firmansya (2018) Mendefinisikan bahwa “PHP (Hypertext Preprocessor) adalah PHP mengizinkan pengembang untuk menempelkan kode didalam HTML dengan menggunakan bahasa yang sama seperti perl dan UNIX shells.”.Sedangkan menurut Isty & Afifah (2018) “PHP adalah pemrograman interpreter yaitu penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”.Menurut pendapat diatas dapat disimpulkan bahwa PHP adalah bahasa pemograman yang digunakan bersamaan dengan HTML dimana bahasa tersebut dijalankan di sisi web browser.

### **2.7.2 HTML**

Menurut Ardhana dalam Kesuma & Rahmawati (2017) mengatakan bahwa.” HTML merupakan suatu bahasa yang dikenali oleh web browser untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video.

### **2.7.3 CSS**

CSS (Cascading Sytle Sheets) banyak dipergunakan untuk memperluas kemampuan HTML dalam memformat dokumen web atau untuk mempercantik tampilan web, bahkan untuk pemosisian dan *layouting* halaman web. Dengan mendefinisikan suatu *style* sekali saja maka style itu akan dapat digunakan berulang kali. (Suyanto, 2009).

### **2.7.4 Xampp**

Menurut Aditya (2011) “XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program: fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang

ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas. Merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis”.

### **2.7.5 Javascript**

Menurut Irawan (2008) “Javascript merupakan bahasa pemrograman berbasis script. Javascript memiliki kemampuan untuk menciptakan halaman web yang dinamis serta didukung oleh banyak web browser.”

Hal ini menjadikan javascript sebagai bahasa script yang paling populer dan banyak digunakan oleh banyak programmer web dalam pengembangan web. Pada aplikasi *client-side*, javascript berjalan didalam kode HTML. Menggunakan javascript kita bisa juga membuat aplikasi interaktif pada halaman web.

## **2.8 Database**

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, penerimaan) barang, hewan, peristiwa, konsep keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. (Andriansyah,2016).

## **2.9 UML( Unified Modeling Language )**

Menurut (Wido,2018) UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat Analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori object oriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna. Tujuan

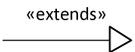
pemodelan dalam kerangka pengembangan sistem adalah sebagai sarana Analisa, pemahaman, visualisasi, dan komunikasi antar tim pengembang yang beranggotakan beberapa/banyak anggota. Beberapa diagram dalam UML yang akan digunakan dalam membantu pengembangan sistem yaitu :

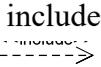
Alat Bantu perancangan sistem yang dapat digunakan adalah :

### 2.9.1 Use Case Diagram

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa dan Shalahuddin, 2016). Adapun simbol-simbol *Use Case* diagram dapat dilihat pada table dibawah ini.

Tabel 2.1 Tabel *Use Case* Diagram

Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>Use Case</i>
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi		Komunikasi antara aktor dan <i>Use Case</i> memiliki interaksi dengan aktor
Ekstensi		Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> , dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman

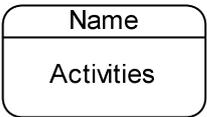
Keterangan	Simbol	Deskripsi
		berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ include/uses	  	<p>Ada dua sudut pandang yang cukup besar mengenai include di <i>Use Case</i> :</p> <p>a. Include berarti <i>Use Case</i> yang ditambahkan akan selalu dipanggil saat <i>Use Case</i> tambahan dijalankan</p> <p>b. Include berarti <i>Use Case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>Use Case</i> yang ditambahkan telah dijalankan sebelum <i>Use Case</i> tambahan dijalankan.</p>

### 2.9.2 Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2016). Simbol-simbol yang terdapat pada activity diagram dapat dilihat pada table dibawah ini.

Tabel 2.2 Tabel Activity Diagram

Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan

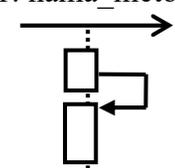
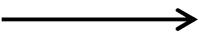
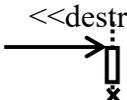
Keterangan	Simbol	Deskripsi
		kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, atau sebuah status akhir.

### 2.9.3 Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *Sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *Use Case* (Salahuddin, 2016). Adapun *symbol sequence diagram* dapat dilihat pada table dibawah ini.

Tabel 2.3 Tabel *Sequence Diagram*

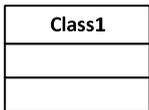
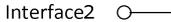
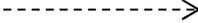
Simbol	Deskripsi
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari
nama actor Atau nama	Aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata

Simbol	Deskripsi
Tanpawaktu aktif	benda di awal frase nama actor
Garishidup/ lifeline 	Menyatakan kehidupan suatu objek
Objek <div style="border: 1px solid black; padding: 2px; width: fit-content;">             nama objek :              nama kelas           </div>	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create 	Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat
Pesan tipe call 1: nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri  1 : nama_metode()
Pesan tipe send 1 : Masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

### 2.9.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2016). Simbol-simbol yang ada pada *class diagram* dapat dilihat dibawah ini.

Tabel 2.4 Tabel *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
atarmuka/interface 	Sama dengan konsep interface dalam pemograman berorientasi objek.
Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole-part).

## 2.10 Metode Pengembangan Sistem

Metodelogi pengembangan sistem pada penelitian ini menggunakan metode *Prototype*. Menurut Adi Nugroho (2011) metode *prototype* dibuat saat pengguna tidak tahu pasti apa yang mereka inginkan baik rincian masukannya, rincian proses dan rincian keluaran yang diinginkan untuk itu dibuatlah *Prototype* kepada pengguna. Kemudian pengguna menyarankan perbaikan-perbaikan jika terdapat kekurangan sistem yang perlu diperbaiki. Adapun tahapan-tahapannya adalah sebagai berikut :

### 1. Interaksi dengan pengguna

Pada tahapan ini penyusun menganalisa apa yang ingin pengguna dapatkan dari sistem/perangkat lunak itu. Sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan sistem.

### 2. Membuat *Prototype*

Pada tahapan ini akan dibuat sebuah *Prototype* aplikasi berbasis web berdasarkan atas kebutuhan pengguna dan sistem pada tahap interaksi dengan pengguna.

### 3. Menguji *Prototype*

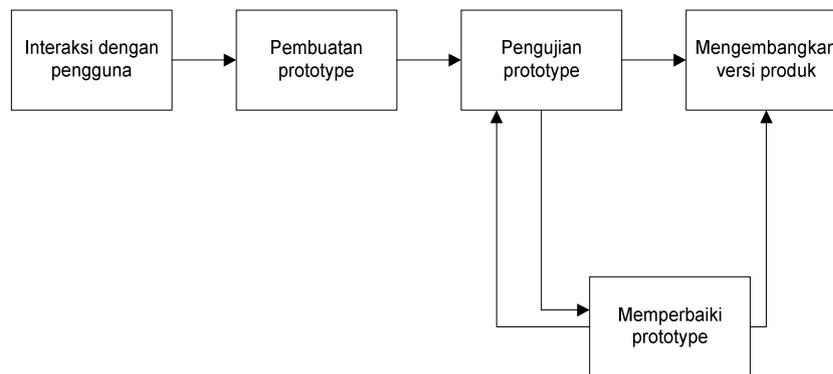
Tahapan ini adalah proses penilaian terhadap *Prototype* yang telah dibuat apakah sesuai dengan kebutuhan atau tidak jika tidak maka *Prototype* akan diperbaiki.

### 4. Memperbaiki *Prototype*

Setelah ditemukan letak kesalahan dari *Prototype* yang dirancang pada tahapan ini penyusun akan membuat atau memperbaiki *Prototype* yang ada setelah itu akan di uji lagi sehingga *Prototype* sesuai dengan keinginan pengguna.

### 5. Mengembangkan Versi Produk setelah aplikasi dapat berjalan dan memenuhi kebutuhan sistem maka aplikasi ini siap dipakai

Metode *Prototype* ini dapat digambarkan sebagai berikut :



**Gambar 2.1** Metode *Prototype* (Adi Nugroho,2011: 78)

Menurut O'Brien (2011), *Prototype* memiliki tahapan tahapan sebagai berikut :

a. Identifikasi kebutuhan Pengguna

Pada tahap ini para pemakai akhir mengidentifikasi kebutuhan bisnis mereka dan menilai kelayakan beberapa alternatif solusi sistem informasi. Pengguna sistem informasi dan vendor mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat beserta alternatif solusi sistem.

b. Membangun *prototype business system*

Pada tahap ini para pemakai akhir atau pakar sistem informasi menggunakan alat pengembangan aplikasi untuk secara interaktif mendesain dan menguji *prototype* berbagai komponen sistem informasi yang memenuhi kebutuhan para pemakai akhir. Membangun *prototyping* aplikasi pengembangan dengan membuat model sebagai uji coba yang mewakili kebutuhan pengguna secara garis besar.

c. Revisi *prototype* kedalam bentuk yang mendekati kebutuhan pengguna  
Model sistem bisnis diuji coba, dievaluasi dan dimodifikasi berulang-ulang hingga dapat diterima oleh pengguna dan dirasakan oleh pengguna telah sesuai dengan kebutuhan.

d. Menggunakan dan memelihara *business system* yang telah diterima

Dalam tahap ini sistem bisnis yang telah disepakati antara pengguna dan *vendor* dapat dimodifikasi dengan mudah karena sebagian besar dokumentasi dari sistem telah tersimpan.

Model *prototype* dimulai dengan pengumpulan kebutuhan. Pengembang dan Pengguna bertemu dan mendefinisikan obyektif keseluruhan dari *software*, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan perencanaan secara cepat. Sehingga *prototyping* dikenal dengan *Rapid Application Development (RAD)*.