



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Aplikasi *Mobile*

Aplikasi *mobile* adalah perangkat lunak tambahan untuk perangkat yang memiliki mobilitas seperti *smartphone* dan *Personal Digital Assistant (PDA)*. Aplikasi populer yang digunakan biasanya adalah *games*, media sosial, berita, dan aplikasi yang berguna untuk kehidupan sehari-harinya. Semua aplikasi itu menggunakan fungsi dari alat komunikasi, audio, video, kamera, sensor, dan *GPS* untuk memaksimalkan kinerja aplikasinya (Alert, 2009)

#### 2.2 *Smartphone*

*Smartphone* menurut *Oxford Dictionary* adalah ponsel yang memiliki banyak fungsi seperti komputer dan memiliki *interface* layar sentuh, akses internet, dan sistem operasi yang mampu menjalankan aplikasi yang di *download* (Oxford, 2018).

#### 2.3 **Komik**

Menurut KBBI **Komik** adalah cerita bergambar (dalam majalah, surat kabar, atau berbentuk buku) yang umumnya mudah dicerna dan lucu. Berdasarkan jurnal

Charles Hatfield yang berjudul *Indiscipline, or, The Condition of Comics Studies* (Hatfield, 2010), syarat sesuatu bisa disebut sebagai komik adalah:

1. Imajinasi yang disusun.
2. Terletak dalam panel.
3. Membentuk Narasi.
4. Terdapat simbol-simbol selain gambar.
5. Susunan panel adalah tuturan khas komik.

Menurut jenisnya, komik bisa dikelompokkan menjadi 2, yaitu comik strip dan comic books. Berikut penjelasannya:

- *Comic Strip*

*Comic Strip* atau komik strip merupakan komik bersambung yang hanya memiliki beberapa panel gambar saja yang umumnya terdiri dari 4 sampai 6 panel. Komik jenis ini menyajikan sebuah alur cerita yang pendek namun tetap dikemas dengan cerita yang utuh.



**Gambar 2.1 Contoh Komik Strip**

**Sumber:** (tahilalats, 2017)

- *Comic Books*

*Comic books* atau komik buku adalah kumpulan cerita bergambar yang terdiri dari satu atau lebih judul dan tema cerita.



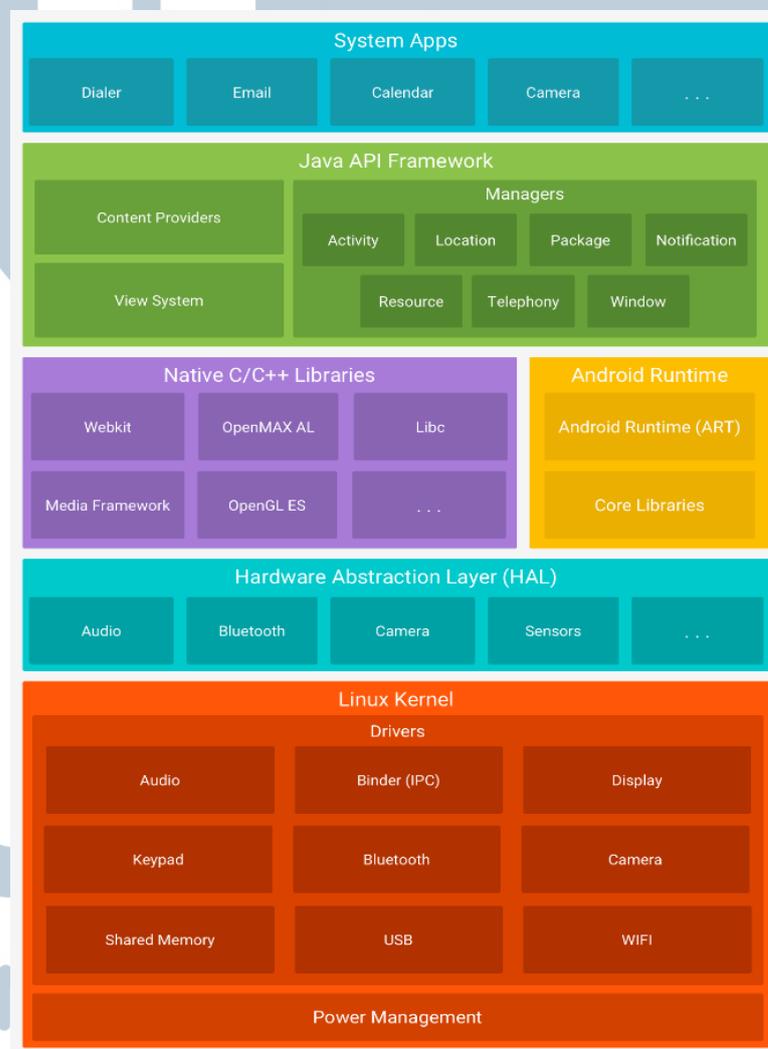
**Gambar 2.2 Contoh Komik Buku**

**Sumber:** (Qbaca, 2018)

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 2.4 Android

*Android* adalah sistem operasi yang berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti smartphone dan computer tablet. Android dapat dipecah menjadi 5 bagian utama, yaitu aplikasi, *framework* aplikasi, *native libraries*, *Android runtime*, dan *Linux Kernel* (Krajci & Cummings, 2013)



**Gambar 2.3** Arsitektur Sistem Android

Sumber: (developer.android, 2018)

## **2.5 User Interface**

*User interface* adalah bentuk tampilan grafis yang berhubungan dengan interaksi antara komputer dengan pengguna. *Interface* pengguna berfungsi untuk menghubungkan antara pengguna dengan sistem operasi (OS) melalui gambar, *icon* dan lainnya, sehingga komputer tersebut bisa digunakan. Dasar pembuatan *user interface* berfokus pada pengguna agar sebuah sistem dapat digunakan dengan mudah (Usability, 2018)

## **2.6 Usability**

Menurut ISO 9241-11 menjelaskan bahwa *usability* menunjuk pada tingkat sebuah produk yang dapat digunakan oleh pengguna tertentu untuk mencapai tujuan spesifik dengan efektif, efisien dan memuaskan dalam sebuah konteks penggunaan. Konteks penggunaan terdiri dari pengguna, tugas, peralatan (hardware, software, dan material), dan lingkungan fisik serta sosial yang mempengaruhi *usability* dalam sistem kerja (ISO, 2018).

## **2.7 Metode Usability Testing**

Usability testing adalah pengujian seberapa mudah desain digunakan pada sekelompok pengguna. Biasanya melakukan pengamatan kepada pengguna saat pengguna berusaha menyelesaikan tugas dan dapat dilakukan untuk berbagai jenis desain. (The Interaction Design Foundation, 2018)

## 2.8 Pengukuran *Usability*

### 2.8.1 *System Usability Scale (SUS)*

*System Usability Scale (SUS)* yang dibuat oleh *John Brooke* pada tahun 1986 adalah metodologi yang andal untuk mengukur *usability*. SUS terdiri dari 10 pertanyaan dengan lima pilihan untuk responden dari sangat setuju hingga sangat tidak setuju. Metode ini memungkinkan untuk mengevaluasi berbagai macam produk dan layanan, termasuk *hardware*, *software*, situs web dan aplikasi (Brooke, 2018) Berikut contoh dari kuisisioner SUS:

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

	Strongly disagree				Strongly agree
1. I think that I would like to use this ballot frequently.	1	2	3	4	5
2. I found the ballot unnecessarily complex.	1	2	3	4	5
3. I thought the ballot was easy to use.	1	2	3	4	5
4. I think that I would need the support of a poll official to be able to use this system.	1	2	3	4	5
5. I found the various parts of this ballot were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this ballot.	1	2	3	4	5
7. I would imagine that most people would learn to use this ballot very quickly.	1	2	3	4	5
8. I found the ballot very awkward to use.	1	2	3	4	5
9. I felt very confident using the ballot.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this ballot.	1	2	3	4	5

**Gambar 2.4** Kuisisioner SUS

**Sumber:** (Brooke, 2018)

U N I V E R S I T A S  
 M U L T I M E D I A  
 N U S A N T A R A

Manfaat menggunakan SUS:

- SUS telah menjadi standar industri, dengan referensi di lebih dari 1.300 artikel dan publikasi.
- Skala yang sangat mudah untuk diberikan kepada peserta.
- Dapat digunakan pada ukuran sampel kecil dengan hasil yang dapat diandalkan.
- Dapat secara efektif membedakan antara sistem yang dapat digunakan dan tidak dapat digunakan.

Penilaian dalam SUS adalah sebagai berikut:

1. Untuk pertanyaan ganji, nilai yang didapat dari responden dikurangi 1.
2. Untuk pertanyaan genap, 5 dikurangi dengan nilai yang didapat dari responden.
3. Skala sangat tidak setuju sampai sangat setuju bernilai 1 sampai 5.
4. Jumlahkan responden yang telah dikonversi dan di kalikan jumlahnya dengan 2,5 agar mendapatkan rentang nilai antara 0-100.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 2.8.2 Penelitian Terdahulu

**Tabel 2.1 Penelitian Terdahulu Yang Berkaitan dengan SUS**

No.	Nama Peneliti	Tahun	Judul Penelitian	Metode Analisis
1	Ika Aprilia H.N, P. Insap Santoso, Ridi Ferdiana	2015	Pengujian <i>Usability Website</i> Menggunakan <i>System Usability</i> <i>Scale</i>	Mengukur tingkat <i>usability website</i> www.tegalkota.go.id menggunakan <i>System Usability</i> <i>Scale</i> .
2	Ardiansyah, Muhammad Imam Ghazali	2016	Pengujian <i>Usability UI</i> dan <i>UX Aplikasi E-Reader</i> Skripsi Berbasis <i>Hypertext</i>	Mengukur akseptabilitas aplikasi <i>E-Reader</i> skripsi berbasis <i>hypertext</i> menggunakan <i>System Usability</i> <i>Scale</i> .
3	Nabilla Ridha Permana, Ismiarta Aknuranda, Retno Indah Rokhmawati	2018	Evaluasi <i>Usability</i> Pada Aplikasi <i>Grab</i> Dengan Menggunakan Metode Pengujian <i>Usability</i>	Mengukur tingkat <i>usability</i> aplikasi <i>Grab</i> dengan menggunakan <i>System Usability</i> <i>Scale</i> .

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

- Pengujian *Usability Website* Menggunakan *System Usability Scale* (H.N, Santoso, & Ferdiana, 2015)

Dalam penelitian ini, dilakukan pengukuran *usability* dari *website* Pemerintah Kota Tegal dengan alamat domain [www.tegalkota.go.id](http://www.tegalkota.go.id) menggunakan SUS (*System Usability Scale*). Kuesioner SUS disebarikan melalui *email* kepada pengguna *website* Pemerintah Kota Tegal yang pernah mengirimkan *email* ke administrator *website*, disebarikan melalui komunitas sosial media masyarakat Tegal, dan juga diedarkan secara langsung ke beberapa kantor Pemerintah. Kuesioner diisi secara online menggunakan *Google Form* atau diisi secara manual. Kuesioner disebarikan selama 1 bulan mulai tanggal 12 Maret 2015 hingga 14 April 2015. Sampel yang digunakan dalam penelitian ini sebanyak 30 orang masyarakat yang menggunakan *website* Kota Tegal.

Hasil yang didapatkan dari penelitian tersebut adalah hasil pengukuran *usability* dapat menjadi langkah awal evaluasi *website*. Skor SUS *website* Pemerintah Kota Tegal sebesar 61.33 yang menunjukkan bahwa *website* belum *usable*, bahkan pengguna berpotensi menjadi *deductor* yang dapat menurunkan jumlah pengguna.

- Pengujian *Usability UI* dan *UX* Aplikasi *E-Reader* Skripsi Berbasis *Hypertext* (Ardiansyah & Ghazali, 2016)

Dalam penelitian ini, dilakukan pengukuran *usability* UI dan *UX* dari aplikasi *E-Reader* skripsi berbasis *hypertext*. Desain aplikasi *eReader* skripsi yang dikembangkan berbasis HTML5. Kinerja *task* diukur menggunakan pengujian

*Post-Task Study* dengan metode *Single Ease Question* (SEQ). Pengujian ini digunakan untuk mengukur kemudahan yang dirasakan pengguna setelah menyelesaikan semua skenario/task yang diberikan sedangkan kinerja *usability* diukur menggunakan metode *System Usability Scale* (SUS).

Penelitian ini berhasil menghasilkan desain *user interface* dan *user experience* aplikasi *E-Reader* skripsi berbasis *hypertext*. Hasil pengujian diperoleh skor SUS sebesar 82.2 menunjukkan bahwa desain yang dibuat memiliki tingkat akseptabilitas yang tinggi sehingga bisa diterima dan siap digunakan oleh pengguna.

- Pengujian *Usability* Aplikasi *Grab* Menggunakan *System Usability Scale*.

Dalam penelitian ini, dilakukan pengukuran *usability* dari aplikasi *Grab* menggunakan SUS (*System Usability Scale*). Karakteristik peserta uji yang digunakan adalah pengguna baru antara 15 sampai 64 tahun dari aplikasi *Grab* dan memiliki pengetahuan mengenai penggunaan *smartphone*. Layanan yang diuji pada aplikasi *Grab* adalah layanan *GrabBike*. Pada layanan *GrabBike* akan dilakukan identifikasi permasalahan dan mengukur tingkat *usability*. Pengukuran tersebut dilihat dari keberhasilan pengguna saat melakukan tugas, lama waktu yang dihabiskan untuk menyelesaikan tugas, dan jumlah kesalahan yang dilakukan ketika mengerjakan tugas.

Ketika pengujian berlangsung aktivitas dari peserta akan direkam dengan menggunakan aplikasi *screen recorder*. Setelah peserta uji menyelesaikan tugas yang diberikan, maka selanjutnya peneliti akan melakukan wawancara. Tujuan dari wawancara adalah untuk mendapatkan data terkait permasalahan yang dialami oleh

peserta ketika menjalankan tugas yang diberikan. Wawancara akan diberikan pada lima peserta yang telah melakukan pengujian. Selanjutnya, peneliti melakukan penyebaran kuesioner pada 100 pengguna aktif dari aplikasi *Grab* dan disebarakan secara *online*

Hasil yang didapatkan dari penelitian tersebut adalah hasil pengukuran tingkat kemudahan pengguna dalam menggunakan aplikasi *Grab* adalah sebesar 100%, tingkat kecepatan pengguna dalam mencari informasi yang dibutuhkan pada aplikasi *Grab* adalah sebesar 0,013, tingkat kesalahan yang dilakukan pengguna terhadap aplikasi *Grab* adalah sebesar 0,13, dan tingkat kepuasan yang dirasakan pengguna aplikasi *Grab* adalah antara 40%-50%. Hasil ini menunjukkan bahwa tingkat kepuasan dari pengguna aplikasi *Grab* memiliki nilai C dimana aplikasi *Grab* masih dibawah rata-rata dan memiliki kemungkinan terdapat masalah dalam aplikasi tersebut.

### **2.8.3 Heuristic Evaluation**

*Heuristic Evaluation* adalah tinjauan ahli atau inspeksi suatu produk menggunakan pedoman tertentu. Dalam penilaian ahli, peninjau sudah mengetahui dan memahaminya. Akibatnya tinjauan ahli cenderung kurang formal dan peninjau tidak perlu menerapkan heuristik spesifik untuk setiap potensi masalah.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 2.9 8 Golden Rules Interface Design

8 Golden Rules menurut Ben Shneiderman pada buku “*Designing the User Interface*” (Shneiderman & Plaisant, 2014) merupakan aturan mendasar dalam perancangan *interface* yang masih sangat bisa digunakan baik dalam *desktop* ataupun *mobile*. Berikut hal-hal yang ada dalam 8 Golden Rules:

### 1. Berusaha Untuk Selalu Konsisten

Konsistensi dibutuhkan setiap halaman dalam suatu aplikasi ataupun antar aplikasi yang masih berhubungan supaya pengguna, terutama pemula, tetap dapat mengenali halaman yang dilihat masih memiliki hubungan dengan aplikasi yang digunakan.

### 2. Memenuhi Kebutuhan Universal

Dalam merancang *interface*, seorang desainer harus memperhatikan variasi penggunaannya. Baik dari segi latar belakang budaya dan bahasa juga variasi tingkat pemahaman user terhadap aplikasi. Pada poin ini yang lebih dipikirkan adalah perbedaan kebudayaan pengguna, sehingga aplikasi harus dirancang dalam berbagai bahasa.

### 3. Menawarkan *Feedback* yang Informatif

*Feedback* yang informatif tidak harus selalu dengan jawaban dari aplikasi ke pengguna, tetapi dapat berupa perubahan *interface* setiap pengguna melakukan aksi. Dengan demikian pengguna paham bahwa aksinya sudah direspon oleh aplikasi.

#### **4. Memberikan Pemberitahuan Jika Suatu Proses Sudah Selesai**

Poin ini termasuk bagian *feedback* yang informatif dengan menyampaikan bahwa suatu proses yang dijalankan sudah selesai.

#### **5. Mencegah Terjadinya Kesalahan**

Poin ini menjaga agar pengguna tidak melakukan kesalahan dalam menjalankan proses. Sangat diperlukan agar pengguna tidak merasa bosan dalam mencoba menggunakan aplikasi karena tidak menemukan aksi yang tepat pada saat mencoba suatu fungsi.

#### **6. Pembatalan Suatu Proses**

Poin ini merupakan salah satu poin yang cukup penting untuk menunjang suatu aplikasi. Yang dimaksud dari poin ini adalah tombol untuk membatalkan suatu proses. Contohnya membatalkan pembelian di *online shop*.

#### **7. *Support Internal Focus of Control***

Yang dimaksud dari poin ini adalah kostumisasi. Pengguna ingin memiliki tampilan yang bisa diatur oleh pengguna sendiri sesuai dengan keinginan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 8. Mengurangi Beban Memori Jangka Pendek

Pada poin ini biasanya desainer berfokus pada desain tata letak menu dan tombol. Tetapi sebenarnya akan lebih efektif jika diterapkan pada saat proses penggunaan harus memberikan input ke sistem. Dengan demikian pengguna tidak perlu mengingat data yang harus di-input ke sistem.

### 2.10 Model Pengembangan *Software*

Proses pengembangan *software* adalah suatu penerapan struktur pada pengembangan suatu *software* yang bertujuan untuk mengembangkan dan memberikan panduan pengembangan sistem. Terdapat beberapa model pengembangan *software*, diantaranya:

#### 1. *Waterfall*

Model *waterfall* adalah suatu proses pengembangan *software* yang dilakukan secara berurutan. Model ini merincikan apa yang seharusnya dilakukan sebelum sistem dikembangkan.

#### 2. *Rapid Application Development (RAD)*

Model *RAD* menekankan pada siklus pengembangan yang singkat dan cepat.

Model *RAD* mengadopsi model *waterfall*.

### 3. *Prototyping*

Model *prototyping* dimulai dengan mengumpulkan kebutuhan pengguna kemudian membuat rancangan yang selanjutnya dievaluasi sebelum diproduksi.

### 4. *Incremental*

Model *incremental* merupakan kombinasi dari model *waterfall*. Model ini dipecah menjadi beberapa bagian sehingga proses pengembangannya bertahap.

### 5. *Spiral*

Model *spiral* adalah model proses software yang merangkai sifat iteratif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial liner

## 2.11 *Unified Modeling Language (UML)*

*Unified modelling language (UML)* adalah bahasa pemodelan visual yang digunakan untuk menentukan, memvisualisasikan, membangun, dan melakukan dokumentasi *software* (Habeeb, 2018). Berikut sekilas tentang 13 diagram dalam struktur diagram UML:

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 1. ***Class Diagram***

*Class diagram* menggambarkan jenis objek dalam sistem dan berbagai macam hubungan yang ada di sistem.

### 2. ***Component Diagram***

*Component diagram* menggambarkan hubungan struktural elemen sistem *software*.

### 3. ***Composite Structure Diagram***

*Composite structure diagram* digunakan untuk menunjukkan struktur internal dari suatu *class*.

### 4. ***Deployment Diagram***

*Deployment diagram* membantu untuk memodelkan aspek fisik dari sistem *software object oriented*.

### 5. ***Object Diagram***

*Object diagram* menunjukkan hubungan antar objek dan menunjukkan bagaimana sistem akan terlihat.

### 6. ***Package Diagram***

*Package diagram* adalah struktur diagram UML yang menunjukkan paket dan dependensi antar paket.

## 7. *Activity Diagram*

*Activity diagram* adalah grafik alur kerja kegiatan untuk menunjukkan aktivitas dari setiap komponen dalam sistem

## 8. *Communication Diagram*

*Communication diagram* digunakan untuk memodelkan perilaku dinamis dari *use case*.

## 9. *Interaction Overview Diagram*

*Interaction overview diagram* berfokus pada gambaran alur interaksi.

## 10. *Sequence Diagram*

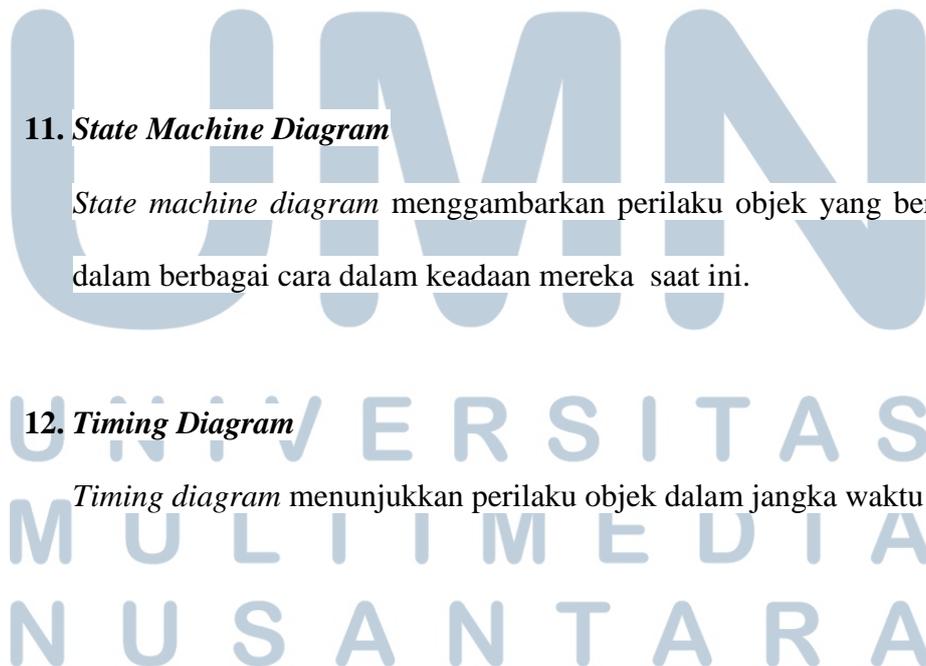
*Sequence diagram* menunjukkan bagaimana objek berinteraksi dengan yang lain dalam skenario tertentu dari *use case*.

## 11. *State Machine Diagram*

*State machine diagram* menggambarkan perilaku objek yang berperilaku dalam berbagai cara dalam keadaan mereka saat ini.

## 12. *Timing Diagram*

*Timing diagram* menunjukkan perilaku objek dalam jangka waktu tertentu.



### **13. Use Case Diagram**

*Use case diagram* menggambarkan bagaimana suatu sistem berinteraksi dengan lingkungannya dengan mengilustrasikan kegiatan yang dilakukan oleh pengguna sistem dan sistem.



## 2.9 Axure RP

*Axure RP* adalah aplikasi *desktop* yang digunakan untuk membuat *wireframe*, purwarupa, *flow diagrams*, dan dokumentasi untuk aplikasi bisnis, *website* dan aplikasi *mobile*. *Axure* mempunyai fitur untuk membuat tampilan visual yang bagus dan memberikan fitur yang mudah dimengerti oleh pengguna yaitu dengan *drag and drop*. *Axure RP* mampu di-ekspor menjadi file HTML.

Berikut adalah fitur-fitur dari *Axure RP* :

1. *Dynamic Content*
2. *Conditional Flow*
3. *Animations*
4. *Data Driven*
5. *Adaptive Views*
6. *Math Function*



**Gambar 2.5** Fitur *Axure RP*

Sumber: (*Axure*, 2018)