



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

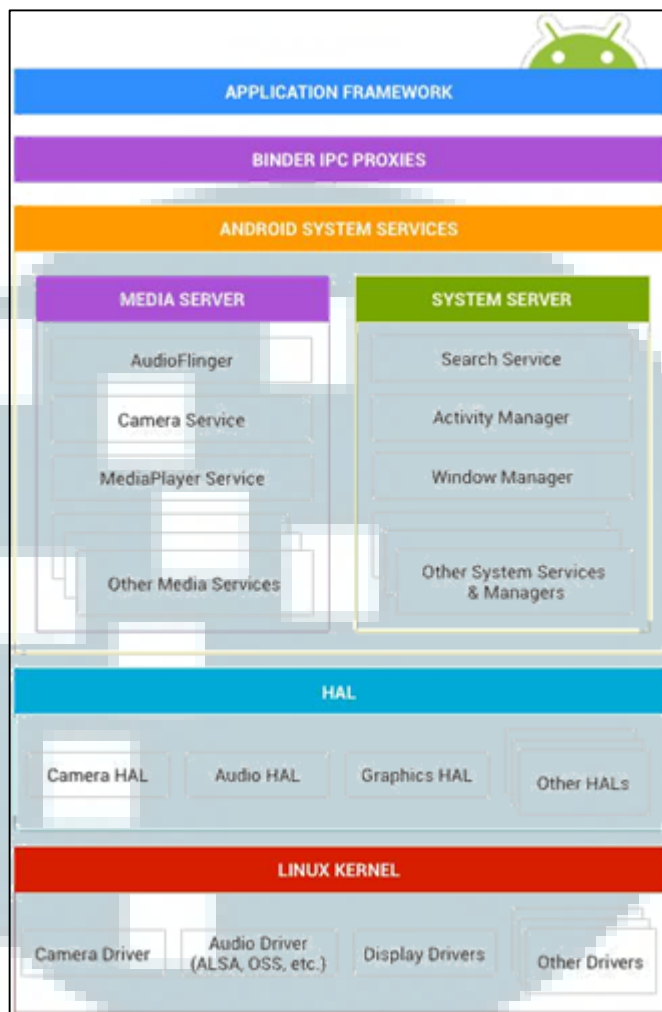
2.1 *Smartphone*

Smartphone atau ponsel cerdas merupakan perangkat yang konsepnya ditentukan pada tahun 1909. Dari konsep itu, *smartphone* pertama dirilis oleh perusahaan Nokia pada tahun 1996 dengan tipe 9000 *Communicator*. *Smartphone* ini sudah dilengkapi dengan aplikasi bisnis dan koneksi internet (Allawy, 2016).

Menurut *Oxford Dictionary*, *smartphone* adalah telpon genggam yang memiliki fungsi seperti computer dan *interface* yang memiliki fitur layer sentuh, akses internet, dan sistem operasi yang memungkinkan untuk menjalankan aplikasi yang sudah didownload.

2.2 **Android**

Android adalah sistem operasi yang *open source* dan memberikan teknologi kepada pengembang dan perusahaan untuk mengembangkan sistem operasi secara independen. Android dapat dipecah menjadi 5 bagian utama, yaitu aplikasi, *framework* aplikasi, *native libraries*, *Android runtime*, dan *Linux Kernel* (Krajci, 2013).



Gambar 2.1 Arsitektur dari sistem Android

Sumber: (Android, 2017)

Gambar 2.1 menunjukkan arsitektur dari Android yang berlaku pada awal pembuatan sampai dengan sekarang.

2.3 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 (International Telecommunication Union, 2009).

2.4 Google Play

Google Play adalah toko aplikasi utama untuk mendistribusikan aplikasi *Android*. Setiap pengembang yang menerbitkan aplikasinya pada *Google Play*, aplikasi tersebut akan diperlihatkan pada 1 milyar pengguna *Android* dalam 190 negara dan daerah di seluruh dunia (Google Inc, 2015).

Menurut Google, ada beberapa alasan kenapa pengembang harus menggunakan *Google Play* sebagai sarana untuk mendistribusikan aplikasi, yaitu sebagai berikut:

- Popularitas yang tinggi dengan pengguna aktif berjumlah lebih dari 1 milyar.
- Performa yang lebih tinggi.
- Fitur tambahan baru
- Pertumbuhan yang pesat.
- Pembayaran ke pengembang yang melebihi 7 milyar *dollar* pada tahun 2014.
- Dari tahun ke tahun, bayaran yang diterima pengembang mengalami peningkatan sebesar 250%
- Lebih dari 50 milyar aplikasi yang sudah didownload dari *Google Play*.

2.5 TransJakarta Busway Navigation

Aplikasi *TransJakarta Busway Navigation* adalah aplikasi yang digunakan untuk mencari rute dari halte asal ke halte tujuan dengan penggunaan *GPS* serta tersedianya *bus tracking*. Aplikasi ini masih menggunakan rute dari tahun 2015 dan akan ditambahkan dengan rute yang tersedia pada tahun 2016. Aplikasi dapat diunduh secara gratis dan hanya sebagai pelengkap mengenai rute tujuan (AsyncByte Software, 2015).



Gambar 2.2 Tampilan dari aplikasi TransJakarta Busway Navigation

Sumber: (AsyncByte Software, 2015)

Gambar 2.2 adalah *screenshot* yang diberikan oleh AsyncByte Software mengenai aplikasi *TransJakarta Busway Navigation*.

2.6 *User interface*

User interface adalah bagian dari perangkat lunak komputer yang orang-orang dapat lihat, dengar, sentuh, dan berbicara. *User interface* memiliki 2 komponen, yaitu *input* dan *output*. *Input* adalah bagaimana pengguna berkomunikasi dengan komputer untuk memenuhi kebutuhan penggunanya. *Output* adalah hasil keluaran yang telah dipenuhi oleh komputer berdasarkan keinginan pengguna (Galitz, 2008).

Menurut Galitz, pentingnya desain yang bagus memberikan pengaruh langsung pada relasi antar organisasi dengan pelanggannya, serta keuntungan yang didapatkan. Desain yang buruk dapat menyebabkan beberapa orang untuk meninggalkan sistem yang digunakan secara permanen dan dapat mempengaruhi tingkat frustrasi orang-orang.

2.7 *Eight Golden Rules*

Eight Golden Rules adalah acuan yang bisa diterima dan sering digunakan oleh murid dan desainer. Dasar ini diambil dari pengalaman dan diperbaharui selama 2 dekade (Shneiderman, 2012). Dasar-dasar dari *Eight Golden Rules* adalah sebagai berikut:

1. Konsistensi

Konsistensi mengarah pada tindakan yang akan dilakukan pada layar yang ditampilkan. Konsistensi juga mengarah pada warna, tampilan, huruf, dan semua yang berhubungan dengan apa yang ditampilkan.

2. *Shorcut*

Menyediakan perintah yang dapat menyelesaikan suatu aksi dengan lebih cepat.

3. Memberikan *feedback* yang informatif.

Setiap pengguna harus mengetahui apa yang mereka lakukan dengan memberikan *feedback* untuk aksi yang dijalankan.

4. Adanya dialog penyelesaian tugas.

Setiap aksi yang dilakukan harus memiliki dialog yang memperlihatkan bahwa pengguna telah selesai dalam melakukan suatu aksi. Setiap langkah dari aksi yang dilakukan harus memiliki dialog yang mengarahkan pengguna.

5. Mencegah *error*.

Desain dari sistem harus memiliki desain yang dapat mencegah *error* yang serius.

6. Memudahkan pembatalan aksi.

Untuk setiap aksi yang dilakukan, harus memiliki media yang dapat membatalkan aksi jika pengguna mengalami kesalahan.

7. Mendukung pengguna mendapatkan kontrol penuh.

Pengguna dapat melakukan kustomisasi terhadap *interface* yang diberikan kepada mereka.

8. Mengurangi beban memori jangka pendek.

Semua tampilan harus terlihat simple agar mudah diingat oleh pengguna. Tampilan yang mudah diingat akan mengurangi beban ingatan jangka pendek pengguna.

2.8 *Usability testing*

Usability testing adalah teknik yang digunakan untuk memastikan bahwa pengguna yang dituju untuk menggunakan sistem dapat mengerjakan tugas secara efisien, efektif, dan memuaskan. *Usability testing* dilakukan pada saat sistem memasuki tahap pra-rilis untuk mengidentifikasi masalah yang terlihat (Gaffney, 2014).

Usability testing mengidentifikasi area dimana orang-orang mengalami kesulitan dengan sebuah produk dan membantu pengembang untuk membuat rekomendasi pada tahap pengembangan. Tujuannya adalah untuk lebih memahami bagaimana pengguna berinteraksi dengan produk pengembang dan ditingkatkan dengan hasil yang diterima (TechSmith, 2008).

Dari 2 definisi tersebut, kesimpulan yang didapatkan adalah *Usability testing* merupakan suatu uji coba untuk mengetahui bahwa sistem yang dibuat dapat digunakan secara layak. Selain penggunaan secara layak, sistem juga harus bisa ditingkatkan berdasarkan informasi yang diberikan oleh peserta yang mengikuti *Usability testing*.

Terdapat 5 komponen utama dalam *usability* (Nielsen, 2012), yaitu:

1. *Learnability* (Kemampuan untuk belajar)

Kemudahan pengguna dalam menyelesaikan suatu kegiatan pada saat pertama kali menggunakan *interface*.

2. *Efficiency* (Efisiensi)

Seberapa cepat pengguna dalam menyelesaikan kegiatan yang diberikan.

3. *Memorability* (Kemampuan untuk mengingat).

Pengguna yang sudah lama tidak menggunakan interface, masih mengingat bagaimana menggunakannya.

4. *Errors* (Kesalahan-kesalahan)

Jumlah kesalahan yang dilakukan oleh pengguna pada saat menggunakan *interface*.

5. *Satisfaction* (Kenyamanan)

Senyaman apa pengguna pada saat menggunakan *interface*.

2.9 Pengukuran Usability

2.9.1 *System Usability Scale (SUS)*

System Usability Scale (SUS) adalah penilaian *usability* yang menggunakan skala *Likert*. *System Usability Scale* mengukur semua komponen utama dari *usability*, yaitu *effectiveness* (kemampuan untuk menyelesaikan tugas menggunakan sistem), *efficiency* (sumber yang dibutuhkan untuk menjalankan tugas), dan *satisfaction* (reaksi subjektif dari pengguna pada saat menggunakan sistem) (Brooke, *SUS - A quick and dirty usability scale*, 1986).

Menurut John Brooke, setiap pernyataan memiliki nilai dari skala 0-4. Pernyataan yang bernomor ganjil dihitung dengan mengurangi posisi skala (skala yang dicentang) dengan 1. Pernyataan nomor genap dihitung dengan

menjadikan posisi skala (skala yang dicentang) sebagai pengurang dari 5. Setelah setiap pernyataan sudah dihitung, semua nilai pernyataan tersebut kemudian dijumlahkan dan mendapatkan nilai total. Kemudian nilai total dikalikan dengan 2,5 untuk mendapatkan nilai *SUS*. Nilai akhir *SUS* didapatkan dari perhitungan rata-rata untuk semua nilai *SUS* yang sudah dihitung.

2.9.2 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu yang berkaitan dengan *SUS*

No.	Nama Peneliti	Tahun	Nama Jurnal	Judul Penelitian	Metode Analisis
1	Camille Peres, Tri Pham, Ronald Philips	2013	<i>Proceedings Of The Human Factors And Ergonomics Society 57th Annual Meeting, 192-196</i>	<i>Validation of The System Usability Scale (SUS): SUS in The Wild</i>	Mengevaluasi 85 partisipan di sekitar kampus dengan <i>SUS</i> untuk mengevaluasi prototipe <i>website</i> , <i>user interface notebook</i>
2	Philip Kortum & Mary Sorner	2015	<i>International Journal of Human-Computer Interaction Volume 31, 518-529</i>	<i>Measuring the Usability of Mobile Applications for Phones and Tablets</i>	Menggunakan <i>SUS</i> untuk mengetahui <i>usability mobile application</i> dari <i>smartphone</i> dan <i>tablet</i> .

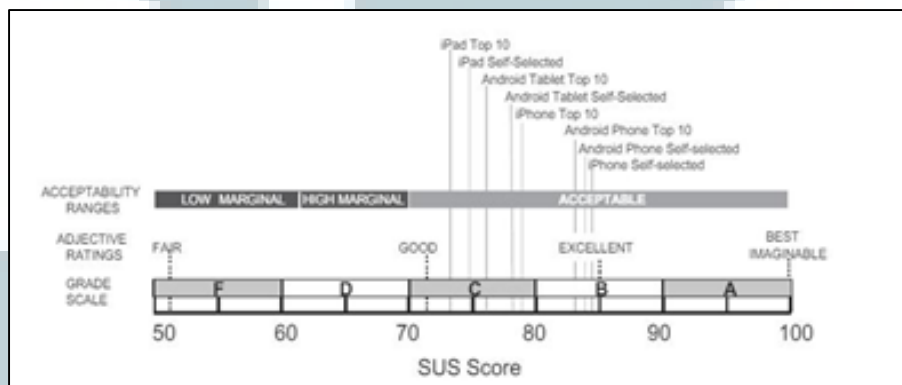
- *Validation of The System Usability Scale (SUS): SUS in The Wild*
(S. Camille Peres, Tri Pham, Ronald Philips)

Dalam penelitian ini, S. Camille Peres membandingkan hasil *SUS* dengan 8 penelitian *usability* yang berbeda. Metode yang digunakan adalah dengan merekrut 85 partisipan di sekitar kampus untuk mengevaluasi *website* yang sudah tersedia, prototipe dari sebuah *website*, dan *user interface* dari *notebook*.

Hasil yang didapatkan dari penelitian tersebut adalah ditemukannya relasi *SUS* dengan 8 penelitian *usability* yang menandakan bahwa *SUS* dapat menjadi pengukuran data yang dapat digunakan untuk membandingkan 2 sistem yang berhubungan dengan *interface* (Peres, Pham, & Philips, 2013).

- *Measuring the Usability of Mobile Applications for Phones and Tablets*
(Philip Kortum and Mary Sorber)

Dalam penelitian ini, dilakukan perbandingan *usability* antara aplikasi pada 2 macam *platform* yaitu *smartphone* dan *tablet*. Penelitian menggunakan sistem operasi iOS dan Android. Data didapatkan dengan memberikan kuesioner *usability* berdasarkan *template* pertanyaan *SUS* kepada 3,575 pengguna. Data untuk penelitian *smartphone* dikumpulkan pada tahun 2013, sementara data untuk penelitian *tablet* dikumpulkan pada tahun 2014.



Gambar 2.3 Hasil skor smartphone dan tablet

Hasil dari penelitian ini menyatakan bahwa aplikasi pada *smartphone* lebih memenuhi konsep *usability* dibandingkan dengan aplikasi yang ada pada *tablet*. Hal ini terlihat dari gambar 2.3 yang memperlihatkan bahwa aplikasi-aplikasi yang ada

pada iOS maupun Android berada pada kategori *acceptable* (dapat diterima) (Kortum & Sorber, 2015).

2.10 Populasi dan Sampel

Populasi merupakan keseluruhan dari orang-orang yang memiliki karakteristik berbeda dan memiliki beberapa bidang untuk diteliti yang dapat digunakan untuk mengambil kesimpulan (Amirullah, 2015).

Sampel adalah sub kelompok dari populasi yang dipilih untuk menjalankan penelitian. Sampel dipilih dengan melewati proses *sampling* atau klasifikasi yang menentukan pemilihan sampel dalam populasi.

Menurut Amirullah, ini adalah beberapa alasan untuk melakukan pengambilan sampel:

- Membatasi jumlah populasi yang terlalu besar
- Pengambilan sampel dapat menghemat biaya
- Pengambilan sampel dapat menghemat waktu yang dibutuhkan
- Pengambilan sampel dapat meningkatkan ketepatan yang lebih tinggi

Untuk menghitung sampel dilakukan perhitungan sebagai berikut:

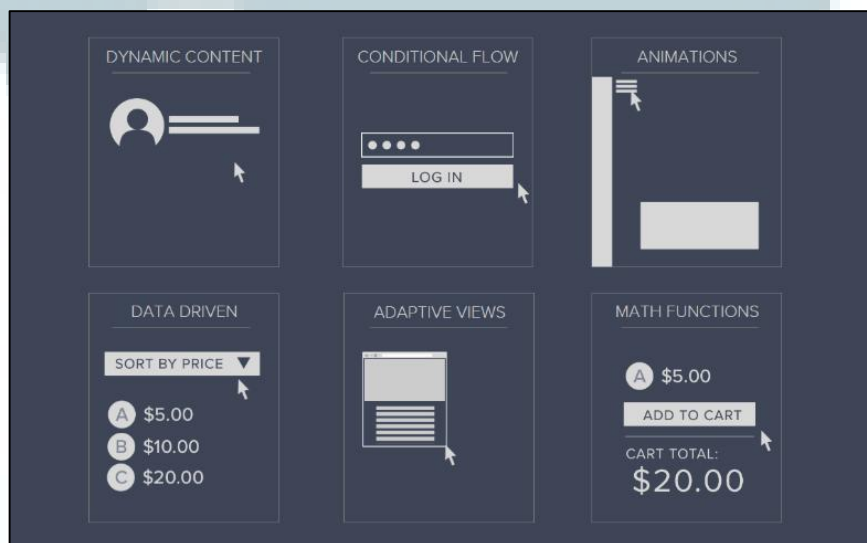
$$\text{Ukuran Sampel} = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + \left(\frac{z^2 \times p(1-p)}{e^2 N}\right)}$$

N adalah jumlah populasi, e merupakan jumlah *error*, dan z adalah jumlah standar deviasi yang digunakan untuk mencari tingkat kepercayaan sesuai dengan jumlah *error* yang diinginkan (Sevilla, 2012).

2.11 Axure RP

Axure RP (Rapid Prototyping) adalah aplikasi yang digunakan untuk membangun sebuah prototipe untuk menemukan solusi dan mengumpulkan *feedback*, melakukan pengulangan pada prototipe sampai mendapatkan solusi, dan mempresentasikan prototipe sebagai bagian dari pengembangan (Victor, 2016).

Fitur-fitur yang diberikan oleh *Axure RP* adalah sebagai berikut:



Gambar 2.4 Fitur-fitur Axure RP

Sumber: (Axure, 2016)

Berdasarkan gambar 2.4, fitur-fitur pada Axure RP adalah sebagai berikut:

- *Dynamic Content*

Axure RP memberikan fitur konten dinamis. Konten dinamis adalah konten yang dapat berinteraksi dengan penggunanya. Contohnya adalah ketika pengguna melakukan *hover* pada suatu gambar, gambar tersebut berubah warna.

- *Conditional Flow*

Conditional flow (alur kondisi) adalah sistem yang ada pada *Axure RP* dimana sistem tersebut menentukan sebuah kondisi itu benar atau salah. Contohnya adalah pada saat pengguna melakukan *login*, sistem akan mengecek apakah *username* dan *password* yang dimasukkan sudah benar atau salah.

- *Animations*

Dalam *Axure RP*, terdapat fitur animasi yang memungkinkan adanya pergerakan yang halus disetiap obyeknya. Contohnya adalah *navigation bar* yang terletak disebelah kiri, yang pada saat diklik akan bergeser ke kanan secara halus.

- *Data Driven*

Data Driven dalam *Axure RP* memberikan fitur yang dapat mengatur *data* yang pengguna masukkan. Contohnya adalah daftar harga yang berbentuk tabel. *Data* yang ada pada tabel tersebut, dapat diurutkan, dirubah, dihapus, atau ditambah.

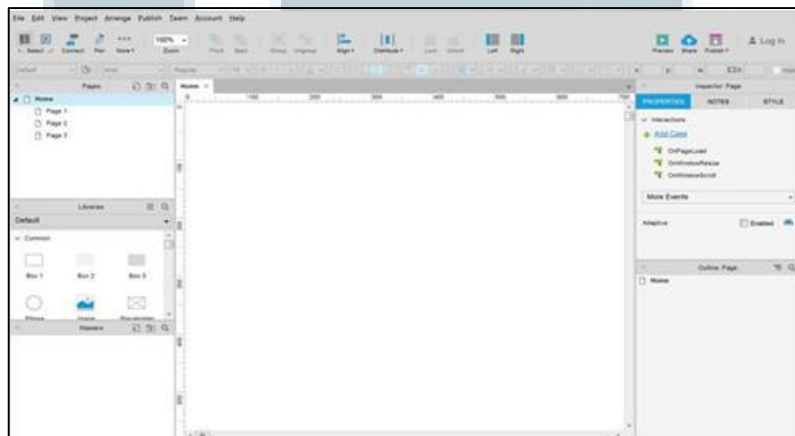
- *Adaptive Views*

Adaptive Views digunakan pada saat pengguna ingin mendesain prototipe dengan berbagai macam ukuran. Contohnya adalah ukuran pada *PC*, *smartphone*, dan *tablet*.

- *Math Functions*

Axure RP memiliki fitur fungsi matematika yang dapat digunakan untuk menghitung nilai dari variabel-variabel yang dimasukkan oleh pengguna. Fitur ini berguna pada saat ingin membuat prototipe dengan keranjang belanja.

Di tengah-tengah *interface Axure RP* terdapat tempat yang dinamakan sebagai *canvas*. Tempat tersebut merupakan daerah dimana *diagram* dibuat dengan meletakkan komponen yang dinamakan sebagai *widget*.



Gambar 2.5 Interface Axure RP

Sumber: (Axure, 2016)

Pada gambar 2.5, terlihat panel bagian atas memiliki fungsi pengaturan spesifik untuk setiap objek yang dipilih. Panel kiri dan kanan digunakan untuk melakukan pengaturan dari segi *user interface* seperti letak obyek atau warna obyek.