



## **Modul : 14**

**CIA – 230 – Analisa Dan Perancangan  
Sistem Informasi**

**Oleh:**

**7841 – Diah Aryani**

**Prodi : Teknik Informatika**

**Designing User Interface**

## A. Kemampuan Akhir Yang Diharapkan

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

Mahasiswa mampu memahami dan menjelaskan *Designing User Interface*

## B. Definisi *Designing User Interface*

### Pengertian User Interface

*User interface* adalah bagian visual dari website, aplikasi software atau device hardware yang memastikan bagaimana seorang user berinteraksi dengan aplikasi atau website tersebut serta bagaimana informasi ditampilkan di layarnya. User interface sendiri menggabungkan konsep desain visual, desain interaksi, dan infrastruktur informasi. Tujuan dari user interface adalah untuk meningkatkan usability dan tentunya user experience.

Antar muka (User Interface) sangat diperlukan dalam suatu pembuata sistem. Dengan adanya antar muka, diharapkan pengguna akan mudah memahami apa yang harus dilakukan dan menghindari terjadinya kebingungan saat menggunakan sistem. Pressman (2010, p312) Desain user interface adalah menciptakan sebuah media komunikasi yang efektif antara manusia dengan komputer.



### PRINSIP UMUM DESAIN USER INTERFACE

Deborah J. Mayhew, dengan *General Principles Of UI Design*, atau Prinsip Umum Desain User Interface. Ada 17 prinsip yang harus dipahami para perancang sistem, terutama untuk mendapatkan hasil maksimal dari tampilan yang dibuat.

- a. **User Compatibility**, yang bisa berarti kesesuaian tampilan dengan tipikal dari user. karena berbeda user bisa jadi kebutuhan tampilannya berbeda. misalnya, jika aplikasi diperuntukkan bagi anak-anak, maka jangan menggunakan istilah atau tampilan orang dewasa.

- b. **Product Compatibility**, istilah ini mengartikan bahwa produk aplikasi yang dihasilkan juga harus sesuai. memiliki tampilan yang sama/serupa. baik untuk user yang awam maupun yang ahli.
- c. **Task Compatibility**, berarti fungsional dari task/tugas yang ada harus sesuai dengan tampilannya. misal untuk pilihan report, orang akan langsung mengartikan akan ditampilkan laporan. sehingga tampilan yang ada bukanlah tipe data (dari sisi pemrogram).
- d. **Work Flow Compatibility**, aplikasi bisa dalam satu tampilan untuk berbagai pekerjaan.. jika tampilan yang ada hanya untuk satu pekerjaan saja. misal untuk kirim mail, maka kita harus membuka tampilan tersendiri untuk daftar alamat.
- e. **Consistency**. Konsisten. Contohnya, jika anda menggunakan istilah save yang berarti simpan, maka gunakan terus istilah tersebut.
- f. **Familiarity**, Icon disket akan lebih familiar jika digunakan untuk perintah menyimpan.
- g. **Simplicity**, aplikasi harus menyediakan pilihan default untuk suatu pekerjaan.
- h. **Direct Manipulation**, manipulasi secara langsung. misalnya untuk mempertebal huruf, cukup dengan ctrl+B.
- i. **Control**, berikan kontrol penuh pada user, tipikal user biasanya tidak mau terlalu banyak aturan.

## 2. DESAIN OUTPUT

Output adalah komponen yang paling dapat dilihat dari sistem informasi yang bekerja/berfungsi. Oleh karena itu, output sering menjadi basis penilaian akhir manajemen terhadap kesuksesan sebuah sistem. Salah satu cara untuk menggolongkan output adalah dengan melihat distribusinya apakah ke dalam atau ke luar perusahaan, dan orang-orang yang membaca dan menggunakan output.

**Internal output** digunakan untuk para pemilik dan pengguna sistem dalam sebuah perusahaan. Output internal mendukung operasi bisnis sehari-hari atau pengawasan manajemen dan pengambilan keputusan.

Tiga jenis output internal adalah sebagai berikut:

1. Detailed Report, menyajikan informasi dengan sedikit atau tanpa dilakukan penyaringan atau pembatasan. Contoh daftar seluruh tagihan pelanggan.
2. Summary Report, berisi informasi dari manajer yang tidak perlu diperlihatkan keseluruhan laporan secara detail. Contoh laporan ringkasan total penjualan dalam hitungan bulanan dan grafik penjualan per-tahun 86
3. Exception Report, menyaring data sebelum ditunjukkan kepada manajer sebagai sebuah informasi. Contoh laporan persediaan barang yang hamper habis.

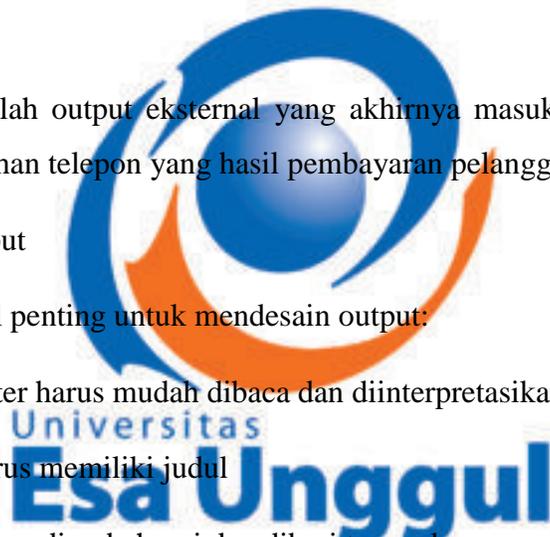
**Eksternal Output** bersifat keluar organisasi. Output ini ditujukan kepada konsumen, pemasok, mitra bisnis dan badan pemerintahan. Output eksternal menyimpulkan dan melaporkan transaksi bisnis. Contoh faktur, nota pembelian, jadwal kursus, tiket pesawat, tagihan telepon dan lain sebagainya.

**Turnaround Output** adalah output eksternal yang akhirnya masuk kembali ke dalam sistem sebagai input. Contoh tagihan telepon yang hasil pembayaran pelanggan menjadi inputnya.

1. Petunjuk Desain Output

Berikut adalah hal-hal penting untuk mendesain output:

1. Output dari komputer harus mudah dibaca dan diinterpretasikan :
  - a. Setiap output harus memiliki judul
  - b. Setiap output harus diperbaharui dan diberi tanggal
  - c. Laporan dan layar (screen) harus memiliki bab dan sub bab pada masing-masing segmen informasinya.
  - d. Pada output berbasis form, semua bagian harus diberi label/nama yang jelas
  - e. Pada output berbasis tabel, semua kolom yang digunakan juga harus diberi nama
  - f. Karena judul bab, field name dan judul kolom sering mengalami penyingkatan untuk menghemat tempat, maka laporan harus menyediakan akses untuk melihat dan menginterpretasikan singkatan tersebut.



- g. Hanya informasi yang dibutuhkan saja yang dicetak atau ditampilkan.
- h. Agar informasi tersebut dapat digunakan, maka informasi tidak boleh diubah secara manual informasi harus tampak seimbang pada laporan atau tampilannya, jangan terlalu penuh atau tersebar.
- j. Para pengguna harus dapat dengan mudah menemukan output tersebut
- k. Jargon komputer dan pesan error harus diabaikan dari semua output

2. Timing output komputer adalah penting.

3. Distribusi atau akses ke output komputer harus mencukupi untuk membantu seluruh pengguna sistem yang relevan.

4. Output komputer harus mudah diterima oleh pengguna sistem yang akan menerima output.

b. Proses Desain Output Langkah-langkahnya adalah sebagai berikut:

1. Mengidentifikasi output sistem dan meninjau persyaratan logis
2. Menentukan persyaratan output fisik

Setelah anda memastikan bahwa anda memahami tipe laporan dari output dan bagaimana output akan digunakan, maka anda perlu menekankan beberapa hal yang terkait dengan masalah desain:

- a. Metode implementasi yang mana yang terbaik untuk melayani output tersebut? Keputusan-keputusan di bawah ini dutuhkan untuk pengguna sistem:
  - Format apa yang paling cocok untuk laporan tersebut? Tabel? Zone? Grafik? Atau gabungan?
  - Jika dibutuhkan printout, anda harus menentukan tipe form atau kertas yang akan digunakan.
  - Pada screen output, anda harus mengetahui batasan peralatan display pengguna.

- Form image dapat disimpan dan dicetak dengan printer laser modern
- b. Seberapa sering output dihasilkan? Sesuai permintaan? Per jam? Harian? Bulanan?  
Untuk output yang sudah terjadwal, kapan pengguna sistem membutuhkan laporan?
  - c. Berapa buah halaman atau lembar dari output yang akan dihasilkan untuk salinan tunggal dari printed output?
  - d. Apakah output membutuhkan banyak salinan? Jika ya, berapa banyak?
  - e. Untuk output yang sudah tercetak, apakah kontrol distribusi sudah terselesaikan? Untuk output online, kontrol akses harus ditetapkan.
3. Mendesain semua preprinted form. Dokumen eksternal dan turnaround dipisahkan untuk pertimbangan tertentu karena mereka berisi informasi yang dinilai konstan dan belum tercetak (preprinted).
  4. Mendesign, memvalidasi dan menguji output. Format atau layout sebuah output secara langsung berpengaruh pada kemudahan pengguna untuk membaca dan 88 menerjemahkannya. Cara paling baik untuk menyusun format tersebut adalah dengan membuat sketsa atau dengan membuat contoh dari dokumen atau laporan.

### 3. DESAIN INPUT

Untuk menginput data ke dalam komputer, analis sistem harus mendesain dokumen sumber, screen input dan metode serta prosedur untuk memasukkan data ke dalam komputer (dari konsumen ke form ke staf entry data ke komputer).

**Data Capture** adalah identifikasi dan penambahan data baru.

**Source document** adalah form yang digunakan untuk menyimpan transaksi perusahaan, khususnya data-data yang ada pada transaksi tersebut.

**Data entry** adalah suatu proses translasi source data atau dokumen ke dalam format yang mudah dibaca oleh computer. Ketika komputasi onlikne menjadi kian umum, maka tanggung jawab data entry sekarang beralih langsung kepada pengguna sistem.

#### a. Masalah Pengguna Sistem pada Desain Input

Input berasal dari sistem, maka human factor memainkan peranan yang sangat penting dalam desain input. Input harus dibuat sesederhana mungkin dan didesain untuk mengurangi kemungkinan kesalahan pemasukan data. Kebutuhan pengguna sistem harus dipertimbangkan.

Jumlah data yang dimasukkan harus seminimal mungkin, semakin banyak data yang dimasukkan, semakin besar potensi kesalahan input dan makin lama juga waktu yang diperlukan untuk menginput data. Jadi, ada berbagai pertimbangan yang perlu dilakukan pada data yang akan di capture sebagai input. Berikut ini prinsip-prinsip desain input yang harus diikuti:

- Dapatkan hanya data variabel, jangan memasukkan data konstan. Misalnya pada input sales order, maka kita membutuhkan part number dari seluruh bagian yang akan dipesan, tetapi kita tidak perlu menginput part descriptions untuk bagian-bagian tersebut.
- Jangan meng-capture data yang dapat dikalkulasi atau dihitung dengan menggunakan program komputer.
- Gunakan kode untuk atribut yang tepat.

Jika source document digunakan untuk menangkap data, maka dokumen tersebut harus mudah digunakan oleh pengguna sistem untuk dilengkapi dan kemudian dimasukkan ke dalam sistem. Beberapa saran di bawah ini dapat membantu :

- Masukkan perintah untuk melengkapi form. Ingat bahwa orang tidak suka jika harus membaca perintah yang dicetak dibalik form.
- Minimalkan jumlah tulisan tangan. Banyaknya orang yang tidak mempunyai kemampuan menulis indah. Petugas data entry dapat salah membaca data lalu salah memasukkan data tersebut.
- Data yang akan dimasukkan harus diurutkan terlebih dahulu sehingga dapat dibaca dari atas ke bawah dan dari kiri ke kanan.

- Jika memungkinkan gunakan desain yang berbasis metapora (input yang mirip dengan kertas) yang sudah dikenal.

**b. Kontrol Internal – Data Editing untuk Input**

Kontrol internal merupakan persyaratan yang ada di seluruh sistem berbasis computer. Control imputer internal menjamin input data pada computer tersebut akurat dan bahwa sistem tersebut aman terhadap suatu kesalahan incidental dan penyalahgunaan.

Di bawah ini petunjuk control internal yang dianjurkan :

- Jumlah input harus diawasi.
- Perhatian juga harus diberikan untuk memastikan bahwa data tersebut valid. Terdapat dua tipe kesalahan yang dapat terjadi pada data: kesalahan data entry dan penyimpanan data invalid oleh pengguna sistem. Kesalahan pemasukan data terdiri dari kesalahan meng-copy, pemindahan posisi (mengetik 123 dengan 123) dan penyelipan (keying 345.36 menjadi 3453.6).

**c. Proses Desain Input**

Langkah-langkahnya adalah sebagai berikut:

- Mengidentifikasi input sistem dan memberikan persyaratan logika
- Memilih control GUI yang sesuai
- Mendesain, memvalidasi dan mengetes input dengan menggunakan beberapa kombinasi dari : Peralatan layout dan Prototyping peralatan.
- Jika perlu, mendesain source document

**4. DESAIN ANTAR MUKA PENGGUNA**

Pada desain antarmuka, audiens adalah system user. System user dapat diklasifikasikan secara luas baik sebagai pakar atau orang baru – dan baik secara terikat dan tidak terikat.

**Expert user (dedicated user)** adalah pengguna komputer yang berpengalaman yang banyak menghabiskan waktunya untuk menggunakan program aplikasi khusus. Expert user

umumnya terbiasa dengan (tetapi tidak perlu ahli dalam) lingkungan operasi aplikasi. Mereka telah menghabiskan waktu untuk belajar menggunakan komputer. Mereka akan menginvestasikan waktu untuk menguasai antarmuka pengguna yang kurang user-friendly. Umumnya, mereka telah hapal operasi rutin dan tingkat di atasnya sehingga tidak memerlukan atau menginginkan feedback atau perintah dari komputer. Mereka ingin dapat mengerjakan tugas mereka dengan tindakan dan keystroke seminimal mungkin.

**Novice user (casual user)** adalah pengguna komputer yang pengalamannya lebih sedikit yang biasanya menggunakan komputer pada frekuensi sedikit atau bahkan pada saat-saat tertentu saja. Boleh dikatakan, novice user membutuhkan lebih banyak bantuan daripada expert user. Bantuan dapat berupa beberapa bentuk, meliputi menu, dialogue, perintah dan help screen.

Kebanyakan sistem yang ada saat ini didesain untuk novice system user, tetapi disesuaikan dengan expert user. Fokusnya adalah user friendliness atau human engineering.

Ahli desain antarmuka pengguna, Wilbert Galitz, mengemukakan masalah masalah antarmuka adalah:

- Terlalu banyak menggunakan jargon atau akronim komputer
- Desain yang tidak jelas atau kurang intuitif
- Tidak mampu membedakan antara tindakan pilihan ("Apa yang harus saya lakukan selanjutnya?")
- Pendekatan pemecahan masalah yang tidak konsisten
- Ketidakkonsistenan desain.

Untuk mengatasi masalah tersebut maka,

- Pahami pengguna anda dan tugas mereka.
- Libatkan pengguna pada desain antarmuka
- Uji sistem pada pengguna aktual. Setelah dilakukan training awal. Amati tindakan dan kesalahan mereka dan dengarkan komentar dan pertanyaan mereka untuk lebih memahami interaksi mereka dengan antarmuka pengguna.

- Lakukan desain interative. Antarmuka pengguna yang pertama mungkin tidak memuaskan. Gunakan desain antarmuka pengguna yang lain untuk melakukan beberapa iterasi desain dan pengujian. Desain antarmuka akan berakhir jika 95% pengguna khusus dapat melakukan tugas yang diharapkan tanpa kesulitan atau bantuan.

#### a. Petunjuk Human Engineering

Dengan tipe pengguna seperti yang telah dijelaskan diatas, beberapa faktor human engineering penting harus digabungkan pada desain:

- Pengguna sistem harus selalu menyadari apa yang harus dilakukan selanjutnya. Sistem harus selalu memberikan perintah tentang bagaimana cara maju, mundur, keluar dan lain-lain. Beberapa situasi membutuhkan tipe feedback:
  - 1) Katakan pada pengguna apa yang sekarang diharapkan oleh sistem. Hal ini dapat dilakukan dalam bentuk pesan sederhana. Contoh “Pilih salah satu”.
  - 2) Katakan pada pengguna bahwa data sudah dimasukkan dengan benar. Hal ini dapat dilakukan sesederhana mungkin, seperti menggerakkan kursor ke field selanjutnya atau menampilkan sebuah pesan seperti “Data OK”.
  - 3) Katakan pada pengguna bahwa data belum dimasukkan dengan benar. Gunakan pesan pendek dan sederhana untuk memberikan pesan tentang kebenaran format.
  - 4) Jelaskan kepada pengguna penyebab penundaan yang terjadi dalam pemrosesan. Contoh pada saat melakukan pencetakan, pengurutan dan sebagainya.
  - 5) Katakan kepada pengguna bahwa tugas telah diselesaikan atau belum diselesaikan. Pesan seperti ”Printing not ready” dan sebagainya.
- Screen harus diformat sehingga bermacam-macam tipe informasi, perintah dan pesan selalu muncul pada area tampilan umum yang sama.
- Pesan, perintah atau informasi harus ditampilkan dengan cukup panjang (secukupnya) sehingga pengguna sistem dapat membacanya.
- Gunakan atribut tampilan dengan hemat. Atribut seperti blinking, higlighting dapat membingungkan jika terlalu banyak.

- Nilai yang salah pada field dan jawaban yang harus dimasukkan oleh pengguna harus ditentukan.
- Antisipasi kesalahan yang dapat dibuat oleh pengguna. Contoh “data akan dihapus?”
- Berkenaan dengan error, pengguna seharusnya tidak diperkenankan untuk meneruskan langkah sebelum memperbaiki error tersebut.
- Jika pengguna melakukan sesuatu yang dapat menimbulkan akibat yang parah, maka keyboard harus dikunci untuk mencegah semua input lain, dan perintah untuk memanggil analis atau technical support harus ditampilkan.

b. Tone dan Terminologi pada Dialogue

Keseluruhan aliran screen dan pesan disebut dialogue. Gunakan kalimat yang sederhana dan benar secara gramatikal, jangan membuat sesuatu yang lucu atau manis dan jangan rendah diri, jangan menghina kecerdasan pengguna sistem.

Berkenaan dengan terminologi yang digunakan pada dialogue komputer, sebaiknya:

- Jangan menggunakan jargon komputer
- Hindari penggunaan singkatan. Jika kita menggunakan singkatan, maka kita menganggap bahwa pengguna telah paham bagaimana menerjemahkannya.
- Gunakan istilah yang sederhana
- Penggunaan terminologi harus konsisten
- Berhati-hati mengungkapkan perintah – gunakan kata kerja tindakan yang tepat.

c. **Pertimbangan khusus untuk Desain Antarmuka**

Pengguna Selain membangun sytle antarmuka pengguna, ada beberapa pertimbangan khusus bagi desain antarmuka pengguna. Bagaimana pengguna akan dikenali dan diautentikasi untuk menggunakan sistem tersebut? Adakah beberapa pertimbangan keamanan atau privasi yang akan diberikan dalam antarmuka pengguna?

Akhirnya bagaimana pengguna akan mendapatkan pertolongan melalui antarmuka pengguna?

### **Internal Control – autentikasi dan otorisasi.**

Pada sebagian besar lingkungan, pengguna harus diautentikasi dan diautorisasi oleh sistem sebelum mereka diizinkan melakukan beberapa tindakan tertentu. Dengan kata lain, pengguna sistem harus "log into" ke dalam sistem. Sebagian besar log-ins membutuhkan **User ID dan Password.**

Terdapat beberapa model untuk membuka dan mengatur sebuah hak istimewa. Satu petunjuk penting adalah menentukan hak istimewa untuk peran/role, bukan untuk individu. Untuk masing-masing peran, perlu ditetapkan hak-hak istimewa khusus yang akan diberikan kepada peran. Hak istimewa tersebut meliputi iin untuk membaca tabel atau view tertentu; izin membuat, mengubah atau menghapus record pada tabel atau view khusus dan sebagainya. View pengguna yang berbeda-beda dapat digunakan untuk mengkustomisasi antarmuka pengguna untuk kategori pengguna yang berbeda-beda. Misalnya cukup mudah untuk "ghost" (mengubah font dari hitam ke abu-abu) dan men-disable opsi menu dan dialogue box yang dilarang untuk beberapa kelompok pengguna sistem.

**Online Help.** Orang menginginkan akses langsung dan segera ke context sensitive help, yakni help yang cukup pintar untuk menggambarkan apa yang dapat mereka lakukan.

**Help system** yang lengkap meliputi daftar isi, berbagai perintah, contoh dan sebuah index yang rinci. Help wizards memandu

### **Fokus Desain Antarmuka**

1. Desain antarmuka Inter-modular;
  - Dikendalikan oleh aliran data antara modul
  - Berkaitan erat dengan desain tingkat komponen
2. Desain antarmuka eksternal

- Antarmuka antar aplikasi
  - Antarmuka antar perangkat lunak dan produsen dan / atau konsumen informasi non-manusia
3. Desain antarmuka manusia-komputer
- Komunikasi antara manusia dan mesin
  - Manusia sebagai pengguna memiliki berbagai karakter yang lebih sulit dipelajari.

### C. LATIHAN

- I. **Sebutkan dan jelaskan Model User Interface**
- II. **Sebutkan Tahapan Analisis dan Perancangan Antarmuka Pengguna**

### D. PEMBAHASAN

- I. Empat model yang berbeda memainkan peranan saat suatu antarmuka pengguna dianalisis dan dirancang:

1) **Pengguna menentukan Model Pengguna**

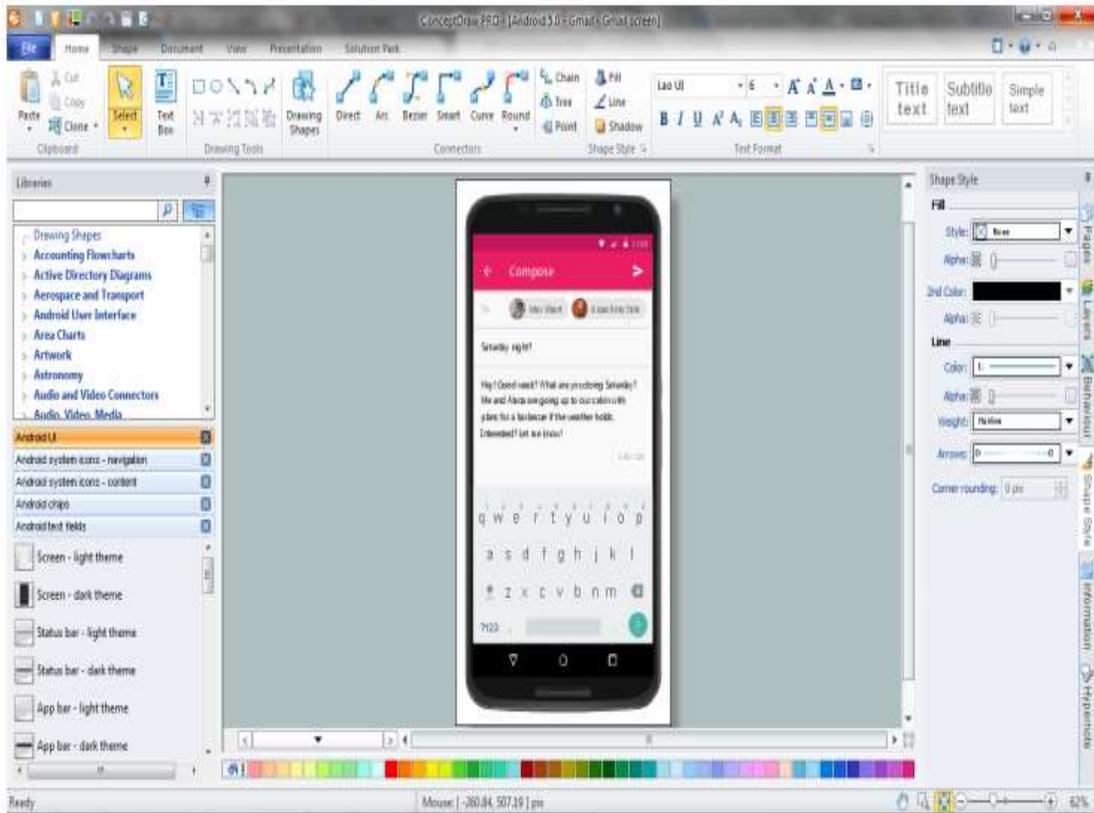
Model Pengguna dibuat untuk mendapatkan profil para pengguna akhir sistem (end user) Pengguna dapat dikelompokkan sebagai berikut:

- Pengguna Pemula
- Pengguna berpengalaman yang tidak terlalu sering menggunakan sistem
- Pengguna berpengalaman yang sering menggunakan sistem

2) **SW engineer membuat Model Desain/Perancangan**

Model desain dari seluruh sistem menggabungkan data, arsitektur, antarmuka, dan representasi prosedural dari perangkat lunak.





### 3) End-user membangun Model Mental atau Persepsi Sistem

- Model Mental (Perspeksi Sistem) adalah gambaran sistem yang ada di kepala para pengguna akhir
- Tingkat ketepatan deskripsi sangat bergantung pada profilprofil pengguna seperti dibahas sebelumnya



#### 4) Pengembang membuat Model Implementasi

- Model Implementasi pada dasarnya menggabungkan manifestasi-manifestasi yang tampak dari luar sistem berbasis komputer, digabungkan dengan informasi-informasi pendukung lainnya (buku-buku, video, dll) yang mendeskripsikan sintak-sintak dan semantik antarmuka pengguna
- Saat model implementasi dan model mental pengguna sesuai, para pengguna secara umum akan merasa nyaman dengan perangkat lunak dan dapat menggunakannya secara efektif

## II. Proses analisa dan perancangan Interface bersifat Iteratif dan dapat direpresentasikan dengan model Spiral

Analisa & Perancangan Interface mencakup 4 aktifitas kerangka kerja:

- User, Task & Environment Analysis (Analisis Antarmuka)
- Interface Design (Perancangan Antarmuka)
- Implementation (Konstruksi Antarmuka)
- Interface Validation (Validasi Antarmuka)

