

Rancang Bangun Aplikasi User Interface Template Berbasis Android

Nola Riska Aprillia Putri¹, Stephanus Eko Wahyudi²

Abstrak— Bagi sebagian *programmer* independen, desain tampilan user interface (antar muka) seringkali dianggap sebagai bagian dari proses desain grafis yang menuntut keahlian dan kreativitas untuk menghasilkan tampilan yang artistik dan lebih fokus pada faktor apakah desain enak dipandang. Banyak *programmer* yang beranggapan bahwa dirinya tidak memiliki kekuatan membuat desain yang estetis atau artistik. Penelitian diawali dengan survei terhadap *programmer* untuk agar mengetahui permasalahan yang dihadapi *programmer* dalam mendesain tampilan antar muka dari aplikasi yang dikembangkan. Hasil survei terhadap 40 responden memberikan konfirmasi terhadap permasalahan tersebut. Hal ini perlu mendapatkan solusi, karena *User Interface* (UI) adalah salah satu bagian terpenting dalam perancangan sebuah aplikasi karena dapat mempengaruhi ketertarikan *user* dalam menggunakannya, terlepas dari banyaknya fitur yang ditawarkan. Berdasarkan masalah tersebut, dikembangkan sebuah aplikasi yang menyediakan *template* UI berdasarkan kategori yang sesuai kebutuhan *programmer* yang memungkinkan untuk menghasilkan luaran berupa *source code* bahasa pemrograman secara otomatis. Aplikasi ini dikembangkan dengan metode Reserve Waterfall dan menggunakan arsitektur Model-View-View Model (MVVM) yang digunakan untuk *live edit* pada tampilan saat adanya perubahan. Hasil dari penelitian ini dapat membantu para *programmer* untuk memilih UI yang diinginkan dan memudahkan pengubahan komponen yang dihasilkan melalui perangkat *mobile*. Kesimpulan dari penelitian ini adalah aplikasi dibuat bermanfaat bagi para *programmer* untuk mempercepat pengembangan aplikasi khususnya dalam proses pembuatan UI kreatif, artistik, dan siap digunakan.

Kata Kunci: Android, User Interface, template UI

Abstract— For some independent programmers, user interface designs are often considered as part of the graphic design process that requires expertise and creativity to produce artistic works. Many programmers think that they do not have the ability to create aesthetic or artistic designs. The research begins with a survey of programmers to find out the problems faced by programmers in designing the application user interface. The results of the survey of 40 programmers as respondents, confirmed that claim. This

problem needs a solution, because the User Interface (UI) is one the most important parts of an application that decide if the user would be interested to use the application, despite many usefull features it offered. Based on these problems, an application was developed that provides UI templates based on categories according to the programmer's needs and generate Java source code automatically. This application was developed using the Reserve Waterfall method and used the Model-View-View Model (MVVM) architecture. The results of this study can help programmers to choose the desired UI and make it easier to change the generated components via mobile devices. The conclusion of this research is that the application is made useful for programmers to accelerate application development, especially in the process of making UI creative, artistic, and ready to use.

Keywords: Android, User Interface, UI templates

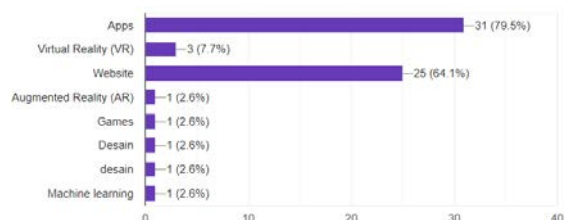
I. PENDAHULUAN

Salah satu permasalahan yang seringkali dihadapi pada *programmer* independen adalah kesulitan pada saat harus merancang tampilan antar muka atau *user interface* (UI). Desain UI seringkali dianggap sebuah proses yang membutuhkan keahlian desain grafis serta kreatifitas untuk menghasilkan karya artistik. Dalam hal ini banyak *programmer* menganggap dirinya memiliki kemampuan berpikir secara logis dan serta kemampuan analisis, namun memiliki kelemahan dalam hal mendesain sesuatu yang artistik. Hal tersebut merupakan permasalahan, karena UI merupakan salah satu bagian penting dalam perancangan aplikasi dan sangat berpengaruh pada minat pengguna untuk memanfaatkannya. UI yang kurang menarik berdampak pada berkurangnya ketertarikan pengguna menggunakan aplikasi, meski banyak fitur yang ditawarkan dan berjalan dengan baik [1].

Langkah pertama yang dilakukan peneliti adalah melakukan survei untuk mengetahui apa yang menjadi permasalahan *programmer* terkait dengan UI. Gambar 1 menunjukkan data hasil survei untuk jenis aplikasi yang banyak dikembangkan responden.

Bidang apa yang Anda kerjakan?

39 responses



Gambar 1. Aplikasi yang dikerjakan programmer

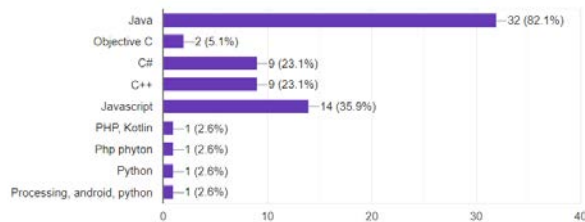
¹ Mahasiswa, Jurusan Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 INDONESIA (telp: 031-745 1699; fax: 031-745 1699; e-mail: nriska@student.ciputra.ac.id)

² Dosen, Jurusan Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 INDONESIA (telp: 031-745 1699; fax: 031-745 1699; e-mail: stephanus.eko@ciputra.ac.id)

Pada pertanyaan kedua yang berupa checkbox untuk pilihan bahasa pemrograman yang paling banyak dipakai para programmer. Terlihat Java dipergunakan 32 dari 40 responden (82,1%) seperti terlihat pada Gambar 2.

Bahasa pemrograman apa yang Anda gunakan?

39 responses

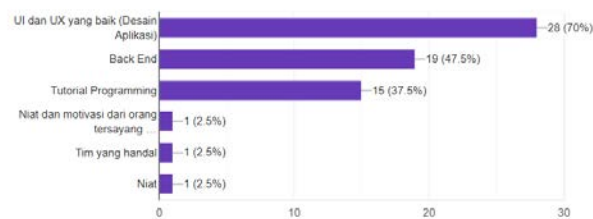


Gambar 2. Bahasa pemrograman yang dipilih programmer

Pertanyaan berikutnya terkait dengan kendala yang dihadapi para programmer. Pada pertanyaan ini sebanyak 28 responden atau sebesar 70% menjawab bahwa pembuatan UI/UX yang baik merupakan kendala yang paling banyak ditemui (Gambar 3).

Sebagai programmer apa lebih menjadi kendala yang Anda alami?

40 responses

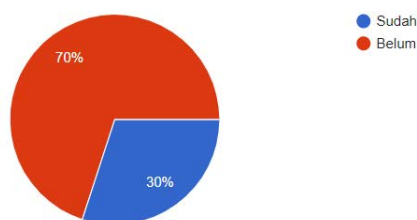


Gambar 3. Kendala yang dialami programmer

Pada saat ditanya apakah programmer sudah menemukan solusi untuk permasalahan yang dihadapi, sebagian besar programmer mengaku belum menemukan solusi terhadap permasalahan yang dialami (Gambar 4).

Dari kendala itu apa sudah ada solusi?

40 responses



Gambar 4. Grafik sudah atau tidak adanya solusi bagi programmer

Hasil dari survei menunjukkan bahwa sebagian besar programmer memilih bahasa pemrograman Java yang dipergunakan untuk mengembangkan aplikasi. Sebanyak 70% responden memiliki problem dalam membuat desain

UI dari aplikasi yang baik, serta belum mendapatkan solusi yang memudahkan permasalahan tersebut.

Berdasarkan permasalahan tersebut, pada penelitian ini akan dikembangkan sebuah alat bantu yang memudahkan programmer untuk mengembangkan UI aplikasi yang memudahkan dalam pengembangan aplikasi. Aplikasi tersebut diharapkan menawarkan berbagai template User Interface (UI) dari berbagai kategori aplikasi. Diharapkan aplikasi menghasilkan *source code* secara otomatis berdasarkan desain yang sudah dipilih dan dikustomisasi. Adapun bahasa pemrograman dari *source code* diharapkan adalah dalam bahasa pemrograman Java untuk dikembangkan melalui Android Studio. Aplikasi yang dikembangkan adalah berbasis mobile pada perangkat yang menggunakan sistem operasi Android.

II. TINJAUAN PUSTAKA

A. User Interface (Antarmuka)

User Interface atau antarmuka pengguna, merupakan tampilan aplikasi yang berupa visual grafis, sebuah fitur yang dimanfaatkan untuk oleh pengguna atau user untuk berhubungan atau berinteraksi dengan software tersebut. Selain itu UI juga dimanfaatkan untuk memudahkan pengguna untuk mendapatkan informasi yang ditampilkan di layar aplikasi. Perancang perlu membangun UI yang baik agar dapat dimanfaatkan agar aplikasi multimedia dapat dimanfaatkan dengan lebih efektif [2]. Fungsi utama dari antarmuka pengguna adalah sebagai penghubung antara pengguna dengan sistem operasi untuk dapat mengoperasikan komputer tersebut [3].

Tujuan pembuatan desain UI yang baik adalah membuat tampilan yang interaktif untuk kebutuhan komunikasi pengguna dengan sistem perangkat lunak [4]. Pengguna sering menilai sebuah sistem bukan dari fungsinya melainkan dari tampilan desain UI yang dimilikinya. Desain UI sebuah aplikasi yang buruk sering menjadi alasan untuk tidak menggunakan perangkat lunak yang ditawarkan. Oleh karena desain UI haruslah memudahkan penggunaan aplikasi, selain membuat aplikasi lebih intuitif dan menarik [5]. Sebuah desain UI harus memiliki sifat yang *user-centered* atau berorientasi pada pengguna, dimana pengguna terlibat dalam proses desain, misalnya melalui survei atau proses evaluasi dengan responden adalah calon pengguna aplikasi terhadap hasil dari desain yang telah dirancang. Selain itu, salah satu studi mengatakan desain UI merupakan hal penting dalam menunjang kehidupan sehari-hari [6].

B. Android

Sistem operasi Android merupakan sistem operasi yang digunakan pada mobile dan tablet agar perangkat dapat dimanfaatkan, memungkinkan interaksi antara pengguna dengan perangkat, serta menjalankan aplikasi-aplikasi yang telah dimuat pada perangkat [7]. Android sendiri dikembangkan oleh konsorsium Open Handset Alliance (OHA) yang merupakan aliansi perusahaan besar

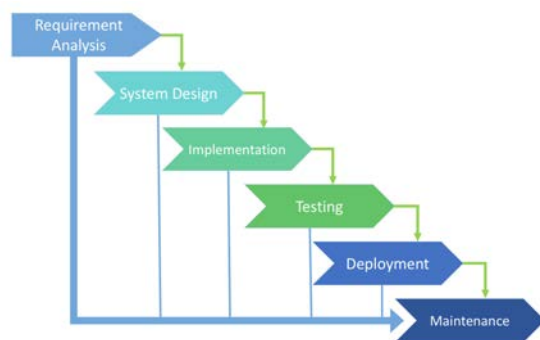
yang memproduksi dan mengoperasikan perangkat keras komunikasi, perangkat lunak dan perusahaan telekomunikasi.

Android merupakan modifikasi dari kernel Linux versi 2.6, dikembangkan oleh Google dengan menggunakan lisensi Apache yang juga artinya merupakan aplikasi *free* dan *open source*. OS ini merupakan kumpulan modul Java dan serta *core library* Java yang berjalan dengan framework aplikasi berorientasi objek dengan basis Java dan Dalvik Virtual Machine (VM). Dalvik merupakan bagian integral dari Android untuk dijalankan di perangkat seluler karena perangkat mobile umumnya memiliki keterbatasan perangkat keras sehingga memiliki keterbatasan kecepatan dan memori prosesor. Untuk dukungan multimedia, OS Android dapat mendukung grafik 2D dan 3D, format audio dan video umum. Ini juga dapat mendukung input multi-sentuh (tergantung pada perangkat) dan membawa peramban Google Chrome V8 JavaScript runtime.

Pada Android, Arsitektur kernel Linux dimodifikasi oleh Google, dimana sudah tidak memanfaatkan X Window System yang asli serta tidak menggunakan dukungan set lengkap dari perpustakaan GNU standar. Hal ini menyebabkan sulitnya untuk melakukan *porting* library atau aplikasi Linux pada Android. Perkembangan Android sendiri telah berjalan dengan sangat pesat, dimana pada tahun 2017 telah dikembangkan 8 versi mulai versi awal Alpha (Versi 1.0) hingga Oreo (Versi 8.0).

C. Metode SDLC Reverse Waterfall

SDLC Reverse adalah salah satu tahapan dalam pembuatan sebuah perangkat lunak yang digunakan dalam pengembangan system-sistem tersebut [8]. Adapun tahapan yang terdapat pada metode ini adalah seperti terlihat pada Gambar 5.



Gambar 5. SDLC Reverse Waterfall Model

Pada penelitian ini aplikasi dikembangkan hingga tahapan Testing dimana dua tahap selanjutnya dikembangkan di kemudian hari. Masing-masing dari empat tahapan tersebut dapat dijelaskan sebagai berikut:

1. Requirement Analysis

Tahap ini merupakan tahapan dimana dilakukan studi melalui jurnal dan publikasi di internet sebagai sumber untuk mendapatkan informasi untuk menentukan

kemungkinan keberhasilan solusi yang akan dibuat. Selain itu untuk mengidentifikasi apakah permasalahan yang ada bisa diselesaikan melalui pengembangan aplikasi yang akan dibuat, serta proses analisis sistem dapat dilakukan dengan melalui survei untuk menentukan kebutuhan sistem secara spesifik yang akan dibuat.

2. System Design

Pada tahapan ini akan dilakukan proses analisis dan desain interaksi dan fungsi pada komponen aplikasi yang akan dibuat, yang kemudian diikuti dengan proses perancangan tampilan dari aplikasi yang akan dibuat.

3. Implementation

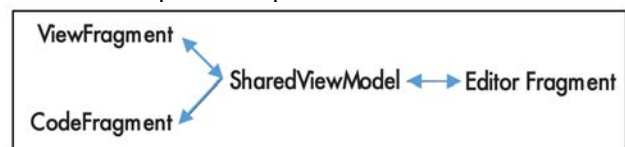
Tahapan ini merupakan tahapan implementasi rancangan berdasarkan desain yang telah dibuat pada tahapan sebelumnya dan perancangan kode atau syntaks.

4. Testing

Pada tahap ini dilakukannya uji coba atau testing apakah aplikasi dapat berjalan dengan baik atau belum. Pengujian dilakukan dengan target pengguna pada aplikasi yang dibuat.

D. Model - View - View Model (MVVM)

Arsitektur MVVM dirancang untuk platform pengembangan UI modern di mana View adalah tanggung jawab seorang desainer daripada pengembang klasik [9]. Pada arsitektur ini terdapat tiga komponen, yaitu Model, View dan View Model. Komponen View adalah terkait dengan UI aplikasi. Arsitektur MVVM lebih ramah dan dapat diterapkan dengan mudah oleh desainer. Komponen kedua adalah Model yang mewakili data. Komponen ketiga dari arsitektur adalah View Model yang terkait dengan data dan operasi untuk melihat dan mengelola logika dan perilaku tampilan [10]. Gambar desain arsitektur dari MVVM ini dapat dilihat pada Gambar 6.



Gambar 6. Desain Arsitektur Model View-View Model

E. Dependent Sample T-test

Metode Dependent Sample T-test ini digunakan saat pengujian dilakukan pada kelompok responden yang sama untuk membandingkan dua kelompok skor dan artinya di mana responden dalam satu kelompok saling berkaitan dengan kelompok lainnya. Salah satu contoh umum dari hubungan semacam itu adalah dalam desain penelitian *post-test* dan *pre-test*. Peserta pada *pre-test* adalah peserta yang sama pada *post-test* dan skor antara *pre* dan *post* test berhubungan secara bermakna. Artinya adalah bahwa nilai antara sebelum dan sesudah ujian bergantung pada satu sama lain. *Post-test* harus diberikan untuk membandingkan skor untuk *pre-test* dan *post-test*. Rumus untuk Dependent Sample T-Test dapat dilihat pada Gambar 7

$$r_{xy} = \frac{N \sum XY - (\sum X)(\sum Y)}{\sqrt{(N \sum X^2 - (\sum X)^2)(N \sum Y^2 - (\sum Y)^2)}}$$

Gambar 7. Rumus Dependent Sample T-test

Keterangan:

N = Jumlah responden

X = Skor yang diberikan oleh rater 1

Y = Skor yang diberikan oleh rater 2

ΣXY = Jumlah perkalian antara variabel X dan variabel Y

III. ANALISIS DAN DESAIN

Pada tahap ini, dilakukannya analisis kebutuhan dari pengguna untuk mengetahui kebutuhan aplikasi terhadap user dalam pengembangan aplikasinya agar dapat menghasilkan solusi yang dapat berguna bagi pengguna. Analisis dilakukan dengan melakukan pengumpulan data melalui kuesioner yang diberikan kepada para programmer.

(1) Pokok Permasalahan

Banyaknya programmer yang merasa menghadapi kesulitan saat merancang UI untuk aplikasi yang mereka kembangkan, biasanya hal ini berawal dari ketakutan atau ketidakpercayaan mereka akan kemampuan dalam mendesain UI. Seperti disebutkan sebelumnya, UI yang kurang menarik akan berdampak pada minat user dalam menggunakan aplikasi meski aplikasi yang ditawarkan sarat akan fitur bermanfaat [1].

(2) Metode

Metode yang digunakan dalam menyelesaikan pokok permasalahan adalah dengan menggunakan dua metode yakni: metode survei untuk mengetahui fitur yang dibutuhkan pengguna dan metode wawancara untuk mengetahui halaman apa saja dari aplikasi yang tampilan UI nya paling sulit untuk didesain.

(3) Kuesioner

a. Tujuan

Tujuan dari survei ini adalah untuk mengetahui fitur yang dibutuhkan pengguna dalam pencapaian solusi yang akan ditawarkan penelitian ini. Solusi dalam mempercepat programmer dalam mendesain sebuah aplikasi.

b. Aspek Pertanyaan

Terdapat 7 pertanyaan yang diberikan kepada para responden. Daftar pertanyaan yang digunakan dalam survei dapat dilihat pada Tabel 1.

TABEL 1
DAFTAR PERTANYAAN SURVEI

No.	Pertanyaan
1	Saya kesulitan dalam mendesain aplikasi mulai dari nol
2	Saya kesulitan dalam menentukan posisi atau letak komponen" (button, textview, dll) dalam

	mendesain UI?
3	Saya kesulitan dalam menentukan besar kecilnya yang preposisi setiap komponen-komponen (button, textview, dll) dalam mendesain UI
4	Saya kesulitan mencari warna yang cocok (skema warna gelap terangnya warna) dalam mendesain UI yang saya mau
5	Saya kesulitan memadukan warna-warna pada color pallete yang cocok (misal biru tua cocok dengan biru muda) pada UI yang saya inginkan
6	Saya kesulitan menentukan bentuk font (misal apps untuk butik maka menggunakan san serif) yang sesuai dengan UI yang saya inginkan
7	Saya kesulitan mendesain UI yang serasi dari page satu dengan page lainnya

c. Responden

Responden dari survei ini adalah para mahasiswa bidang Teknik Informatika Universitas di wilayah kota Surabaya yang berpengalaman membuat aplikasi berbasis Android. Dari survei yang dilakukan telah mendapatkan sebanyak 43 responden.

d. Hasil Survei

Berdasarkan survei yang telah dilakukan terdapat hasil survei dapat dilihat pada Tabel 2.

TABEL 2
DAFTAR PERTANYAAN DAN HASIL SURVEI

No.	Pertanyaan	SS	S	N	TS	STS
1	Saya kesulitan dalam mendesain aplikasi mulai dari nol	0	7,3	22	29,3	41,5
2	Saya kesulitan dalam menentukan posisi atau letak komponen" (button, textview, dll) dalam mendesain UI?	4,9	19,5	29,3	22	24,4
3	Saya kesulitan dalam menentukan besar kecilnya yang preposisi setiap komponen-	4,9	2,4	19,5	34,1	39

	komponen (<i>button</i> , <i>textview</i> , dll) dalam mendesain UI					
4	Saya kesulitan mencari warna yang cocok (skema warna gelap terangnya warna) dalam mendesain UI yang saya mau	0	4,9	14,6	26,8	53,7
5	Saya kesulitan memadukan warna-warna pada <i>color pallete</i> yang cocok (misal biru tua cocok dengan biru muda) pada UI yang saya inginkan	0	2,4	14,6	22	61
6	Saya kesulitan menentukan bentuk <i>font</i> (misal <i>apps</i> untuk butik maka menggunakan <i>san serif</i>) yang sesuai dengan UI yang saya inginkan	2,4	4,9	22	22	48,8
7	Saya kesulitan mendesain UI yang serasi dari <i>page</i> satu dengan <i>page</i> lainnya	2,4	9,8	19,5	19,5	48,8

Keterangan:

SS: Sangat Setuju

S: Setuju

N: Netral

TS: Tidak Setuju

STS: Sangat Tidak Setuju

e. Analisis Survei

Dari hasil survei tersebut dapat disimpulkan masalah yang perlu diselesaikan untuk memenuhi

kebutuhan programmer agar mempercepat proses desain aplikasi adalah sebagai berikut:

- 1) Dalam mendesain aplikasi butuh waktu lebih lama
- 2) Sulit dalam mendesain aplikasi mulai dari nol
- 3) Sulit menentukan preposisi yang baik untuk tata letak komponen-komponen UI
- 4) Sulit dalam menentukan besar kecilnya yang preposisi setiap komponen-komponen (*button*, *textview*, dll) dalam mendesain UI
- 5) Sulit mencari warna yang cocok (skema warna gelap terangnya warna) dalam mendesain UI yang pengguna mau
- 6) Sulit memadukan warna-warna pada *color pallete* yang cocok (misal biru tua cocok dengan biru muda) pada UI yang pengguna inginkan
- 7) Sulit menentukan bentuk font yang sesuai dengan UI yang pengguna inginkan
- 8) Sulit mendesain UI yang serasi dari *page* satu dengan *page* lainnya.

(4) Wawancara

Wawancara dilakukan untuk mengetahui tampilan apa saja yang menjadi kesulitan programmer. Hasilnya akan dimanfaatkan untuk menentukan kategori desain pada aplikasi.

a. Tujuan

Tujuan dari survei ini adalah untuk mengetahui fitur yang dibutuhkan pengguna dalam pencapaian solusi yang akan ditawarkan penelitian ini. Solusi dalam mempercepat programmer dalam mendesain sebuah aplikasi.

b. Aspek Pertanyaan

Terdapat 3 pertanyaan yang diberikan kepada para responden. Daftar pertanyaan yang digunakan dalam survei dapat dilihat pada Tabel 3.

TABEL 3
DAFTAR PERTANYAAN WAWANCARA

No.	Pertanyaan
1	Tampilan halaman apa yang paling sulit dalam membuat desain tampilan sebuah aplikasi?
2	Mengapa sulit dalam mendesain tampilan <i>page</i> tersebut?
3	Jenis aplikasi apa yang dibuat dan butuh waktu lama saat mendesain tampilan aplikasinya? Mengapa?

c. Responden

Wawancara ini dilakukan terhadap 5 programmer dengan tujuan untuk mengetahui kebutuhan pengguna dalam kategori desain aplikasi bagian apa yang menjadi hal yang sulit saat programmer mendesain sebuah aplikasi. Untuk daftar nama programmer yang diwawancarai dapat dilihat pada Tabel 4.

TABEL 4
DAFTAR PROGRAMMER YANG DIWAWANCARAI

No	Nama	Universitas
1	Jefferson Phinardi Kosasih	Universitas Ciputra
2	Kevin Hartarto	Universitas Ciputra
3	Ricardo Ryan	STTS Surabaya
4	Ardian Firman	Ubaya Surabaya
5	Andre	Petra Surabaya

d. Hasil Wawancara

Hasil dari wawancara, sebagai berikut:

i. Tampilan halaman apa yang paling sulit dalam membuat desain tampilan sebuah aplikasi?

Pada pembahasan pertanyaan ini, dari kelima programmer mengatakan bahwa halaman yang paling sulit untuk dipikirkan bentuk desain yang menarik adalah login, register dan profile.

ii. Mengapa sulit dalam mendesain tampilan page tersebut?

Dari pertanyaan sebelumnya terjawab bahwa kesulitan terjadi paling sulit saat mendesain page Login, Register dan Profile, hal itu dikarenakan bagi programmer page tersebut berisikan hanya terdapat sedikit jenis komponen dan akan beresiko adanya ruang kosong yang banyak dan terlihat kurang bagus namun mereka bingung bagaimana agar komponen-komponen tertata dengan indah dan enak dipandang.

iii. Jenis aplikasi apa yang dibuat dan butuh waktu lama saat mendesain tampilan aplikasinya? Mengapa?

Bagi lima programmer yang diwawancari, mendesain aplikasi *marketplace* dan *social media* merupakan yang paling lama karena dibutuhkan desain yang menarik untuk benar-benar dapat menarik perhatian pengguna. Oleh karena itu bagian ini memerlukan perhatian khusus, misalnya untuk menentukan warna yang cocok, besar kecilnya font dan jenis font yang cocok, lalu konten-konten yang perlu ditambahkan seperti logo, text dan lain-lainnya. Seringkali hasil yang dibuat kurang memuaskan programmer karena terlalu sederhana dan polos, namun programmer sudah tidak tahu langkah berikutnya yang harus dilakukan.

e. Analisis Wawancara

Berdasarkan hasil wawancara yang dijelaskan pada subbab sebelumnya dapat dianalisis bahwa pengguna mengalami kesulitan mendesain aplikasi pada halaman login, register dan profile. Hal ini dikarenakan secara umum terdapat komponen yang sedikit sehingga terdapat ruang kosong serta terlihat monoton. Selain itu, aplikasi *marketplace* dan *social media* merupakan contoh aplikasi untuk didesain oleh programmer.

f. Solusi yang Ditawarkan

Berdasarkan permasalahan yang didapat dari hasil survei, berikut beberapa fitur yang ditawarkan sebagai solusi untuk menyelesaikan masalah tersebut Tabel 5.

TABEL 5
SOLUSI YANG DITAWARKAN

No	Masalah	Solusi
1	Sulit dalam mendesain aplikasi mulai dari nol	Fitur yang menyediakan daftar template UI dan daftar template UI sesuai kategori
2	Sulit menentukan preposisi yang baik untuk tata letak komponen UI	Membuat fitur edit posisi (margin) komponen
3	Sulit dalam menentukan besar kecilnya yang preposisi setiap komponen (button, textview, dll) dalam mendesain UI	Fitur edit size pada komponen yang ingin diubah ukurannya
4	Sulit mencari warna yang cocok (skema warna gelap terangnya warna) dalam mendesain UI yang pengguna mau	Fitur edit warna yang direkomendasikan aplikasi
5	Sulit memadukan warna-warna pada color pallete yang cocok (misal biru tua cocok dengan biru muda) pada UI yang pengguna inginkan	Fitur edit warna yang sudah terpadu yang direkomendasikan aplikasi
6	Sulit menentukan bentuk font yang sesuai dengan UI yang pengguna inginkan	Fitur pemilihan font pada template UI yang diinginkan dengan fontstyle yang direkomendasikan aplikasi
7	Sulit mendesain UI yang serasi dari page satu dengan page lainnya	Fitur kategori pada jenis-jenis UI yang menyediakan template UI yang sesuai dengan beberapa page

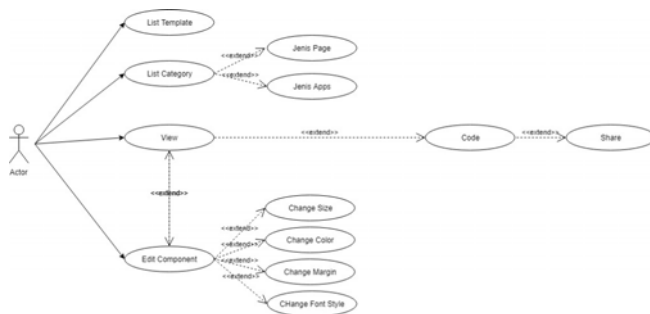
g. Kebutuhan Aplikasi

Fitur-fitur tersebut akan dikembangkan dalam bentuk aplikasi mobile yang dapat dijalankan pada sistem operasi Android dengan spesifikasi sebagai berikut:

- Sistem operasi minimal Android versi 4.0
- Memiliki akses Internet memadai

(5) Use case Diagram

Aplikasi ini dioperasikan dengan beberapa aksi yang dapat dilakukan user dengan tujuan untuk mencari template UI yang dapat digunakan dalam pengembangan aplikasinya. User dapat melakukan aksi yang dapat dilihat dari Gambar 10.



Gambar 1. Use Case Diagram

Fitur-fitur ini dijelaskan sebagai berikut.

1. List Template
User dapat melihat berbagai macam template UI yang tersedia pada aplikasi.
2. List Category
User dapat memilih kategori UI sesuai jenis page maupun jenis tema UI yang dicari.
3. Page Category
User dapat memilih kategori berdasarkan Page Category.
4. Apps Category
User dapat memilih kategori berdasarkan Apps Category.
5. View
User dapat melihat tampilan UI yang dipilih secara fullscreen.
6. Edit Component
User dapat mengubah setiap komponen seperti yang diinginkan.
7. Change Size
User dapat mengubah ukuran komponen pada template UI sesuai keinginan.
8. Change Color
User dapat mengubah warna komponen pada template UI sesuai keinginan.
9. Change Margin
User dapat mengubah ukuran margin pada komponen pada template UI sesuai keinginan.
10. Change Font Style
User dapat mengubah jenis font pada komponen pada template UI sesuai keinginan.
11. Code
User dapat melihat source code UI yang dipilih.
12. Share
User dapat membagikan atau mengirimkan source code file melalui email.

Pada fitur List Template, user membuka aplikasi maka sistem akan menampilkan page dashboard dimana akan ditampilkan list template UI atau daftar-daftar UI

yang disediakan oleh aplikasi. Lalu user dapat melihat macam-macam template User Interface. Jalannya fitur List Template ini dapat dilihat pada Gambar 8.

Pengguna dapat memilih kategori yang diinginkan dengan membuka aplikasi maka sistem akan menampilkan kategori-kategori UI yang dapat dipilih. Lalu user dapat melihat macam-macam template User Interface sesuai kategori. Jalannya fitur List Category dapat dilihat pada Gambar 9.

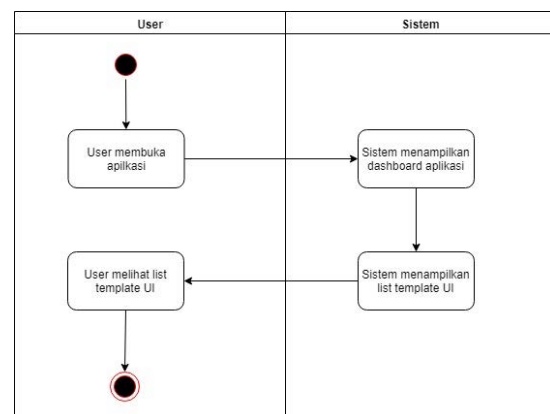
Setelah user melihat kategori yang ditampilkan oleh sistem lalu user memilih template UI berdasarkan jenis page maka sistem akan menampilkan template UI yang berdasarkan jenis page dan apps saja. Lalu user memilih template yang diinginkan. Jalannya fitur Page Category dan Apps Category ini dapat dilihat pada Gambar 10 dan 11.

Kemudian user memilih template UI yang diinginkan maka sistem akan menampilkan tampilan UI yang dapat dilihat sesuai dengan ukuran layar Android smartphone. Jalannya fitur View ini dapat dilihat pada Gambar 12.

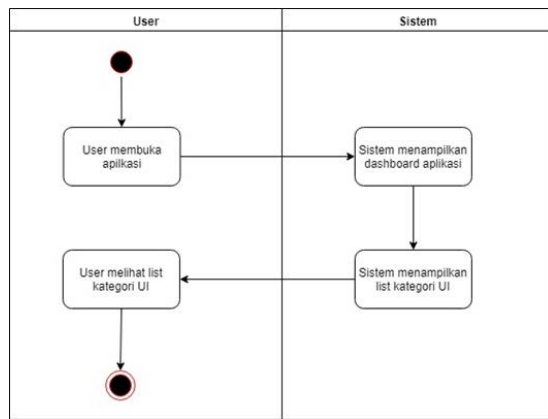
Saat sistem menampilkan view UI yang dapat dilihat. Lalu user memilih komponen mana yang ingin diubah maka sistem akan menampilkan atau memunculkan *pop up* pengeditan. Setelah user melakukan pengeditan, user melakukan penyimpanan dengan menekan tombol *apply*. Pada fitur ini terdapat fitur change size, change color, change margin, change font style. Jalannya fitur Edit Component ini dapat dilihat pada Gambar 13.

Proses fitur code berguna saat user memilih template desain aplikasi yang diinginkan maka secara otomatis akan memunculkan code pada tab code. Jalannya fitur Code ini dapat dilihat pada Gambar 14.

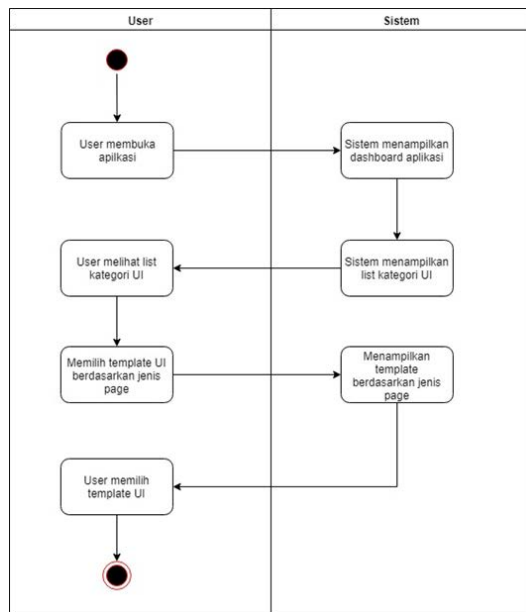
Setelah user memilih template UI yang diinginkan maka sistem akan menampilkan code UI yang dapat dilihat lalu user dapat membagikan code tersebut melalui email. Jalannya fitur Code ini dapat dilihat pada Gambar 15.



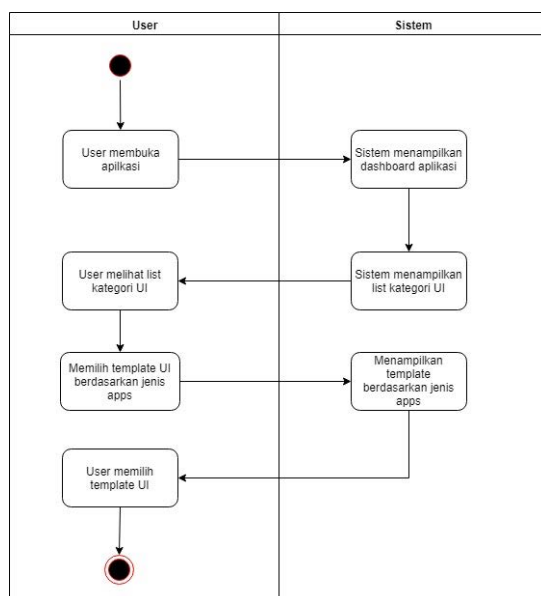
Gambar 8. Activity Diagram untuk List Template



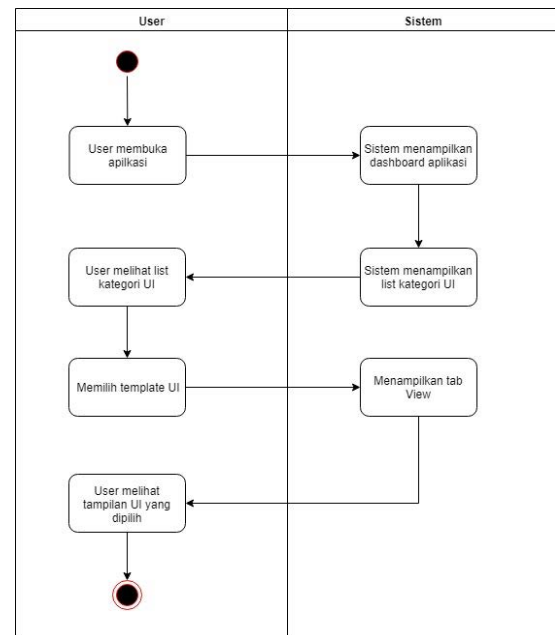
Gambar 9. Activity Diagram untuk List Category



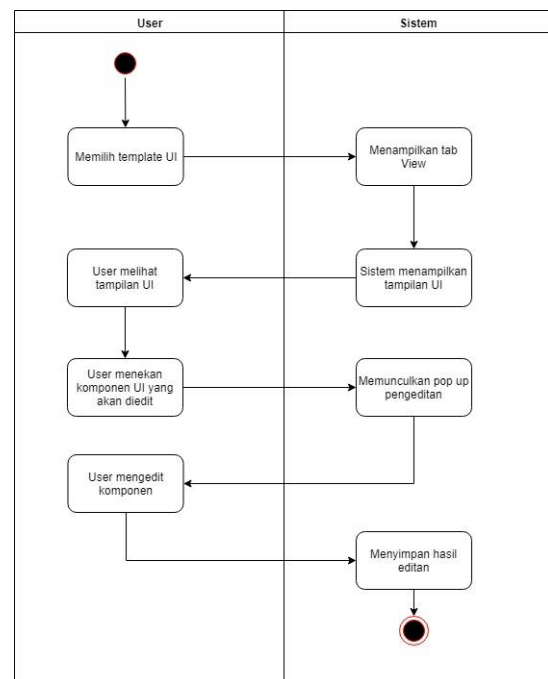
Gambar 10. Activity Diagram untuk Page Category



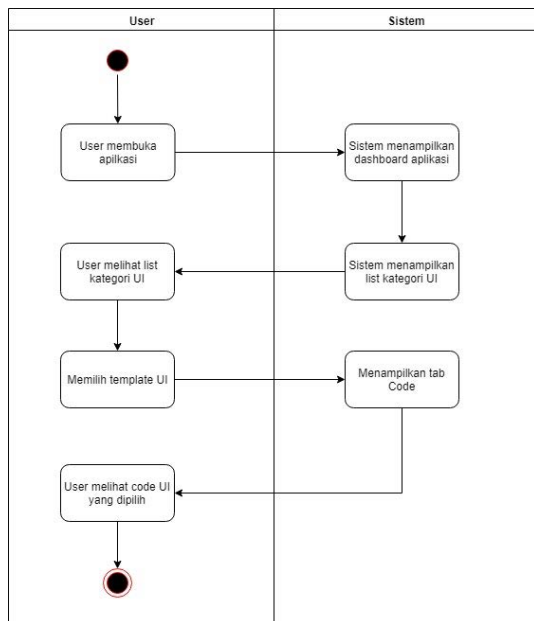
Gambar 11. Activity Diagram untuk Apps Category



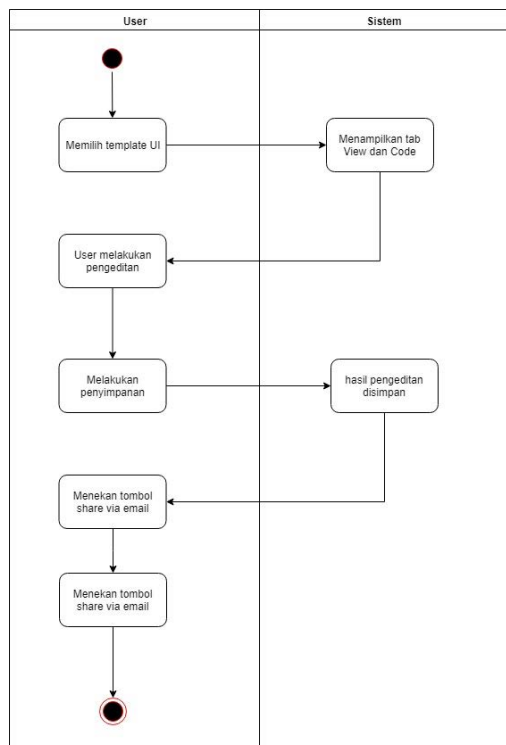
Gambar 12. Activity Diagram untuk View



Gambar 13. Activity Diagram untuk Edit Component



Gambar 14. Activity Diagram untuk Code

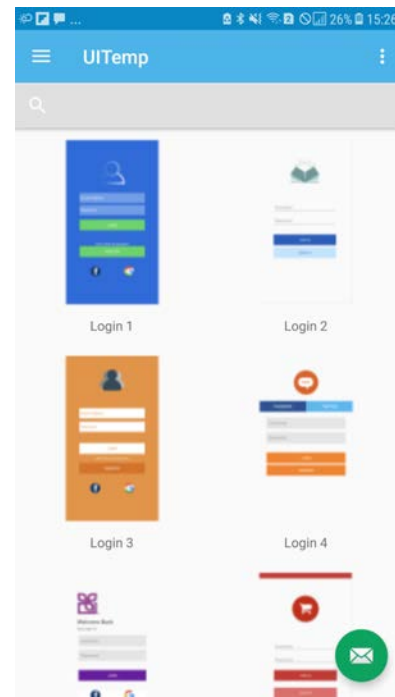


Gambar 15. Activity Diagram untuk Share

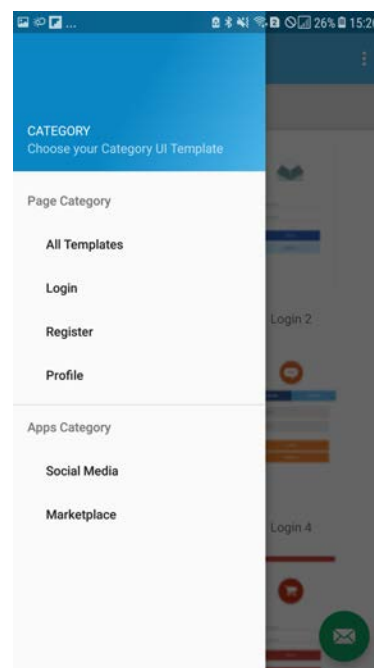
IV.IMPLEMENTASI

Beberapa tampilan hasil dari implementasi dapat dilihat pada Gambar 16 hingga Gambar 23. Pengguna dapat melihat dan memilih template seperti pada Gambar 16. Pengguna dapat memilih kategori yang tersedia seperti pada Gambar 17. Kategori yang dipilih pengguna dapat memilih berdasarkan jenis *page* dan *apps category* seperti pada Gambar 18 dan 19. Pengguna dapat melihat tampilan template secara penuh seperti Gambar 20. Tampilan pop up

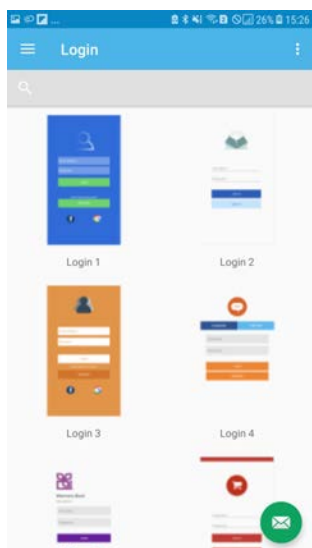
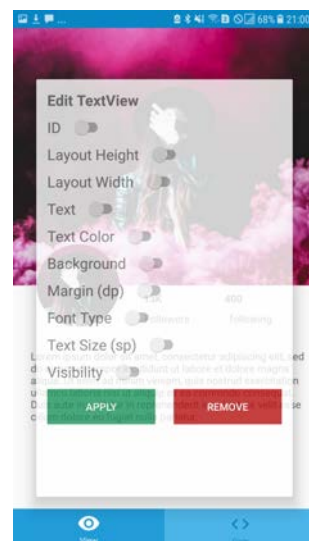
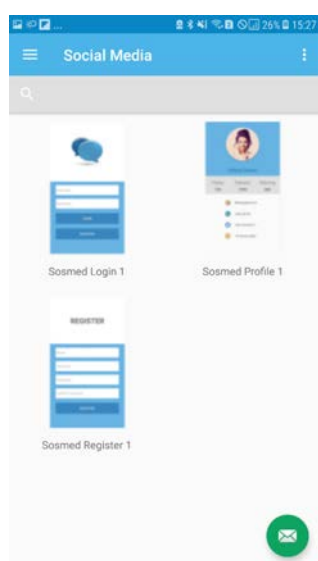
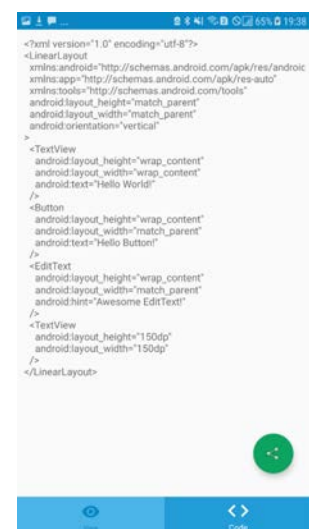
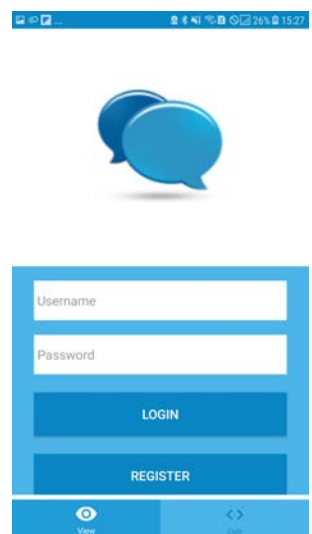
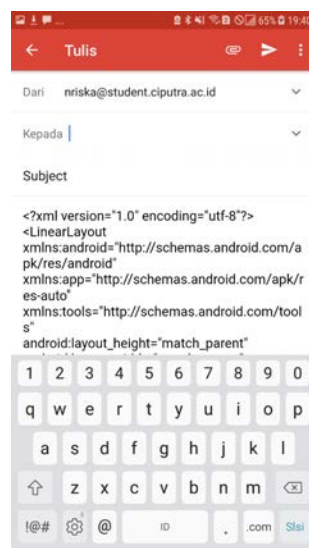
yang berfungsi untuk pengeditan seperti pada Gambar 21. Pada tampilan pengeditan dapat Change Size, Change Color, Change Margin dan Change Font Style. Tampilan fitur code dimana user akan mendapatkan source code dari tampilan template UI yang dipilih dapat dilihat pada Gambar 22 serta pada Gambar 23 merupakan tampilan fitur share dimana pengguna dapat membagikan source code tersebut.



Gambar 16. Halaman List Template



Gambar 17. Halaman List Category

Gambar 18. Halaman *List Page Category*Gambar 21. Halaman *Edit Component*Gambar 19. Halaman *List Apps Category*Gambar 22. Halaman *Code*Gambar 20. Halaman *View*Gambar 23. Halaman *Share*

Setelah tahap implementasi selesai dilakukan, maka selanjutnya dilakukan tahap pengujian untuk mengetahui apakah aplikasi ini menyelesaikan masalah yaitu untuk mempercepat para programmer dalam mendesain aplikasi. Pengujian manfaat atau validasi ini dilakukan dengan tujuan mengetahui seberapa bermanfaatnya aplikasi bagi pengguna yakni para programmer yang berpengalaman dengan platform Android. Pengujian ini adalah lanjutan dari pengujian fitur dan dilakukan terhadap responden yang sama. Kemudian responden diminta untuk mengisi form survei yang diberikan secara *print out*.

Dalam pengujian ini, pengambilan data menggunakan metode survei simple random sampling. Responden menjawab pertanyaan dengan menggunakan skala likert yaitu skala 1 hingga 5 dengan ketentuan sebagai berikut.

- 1 untuk Sangat Tidak Setuju
- 2 untuk Tidak Setuju
- 3 untuk Netral
- 4 untuk Setuju
- 5 untuk Sangat Setuju

Survei dilakukan sebelum dan sesudah melakukan aplikasi guna untuk mengetahui pengaruh aplikasi terhadap pengguna dengan menggunakan metode Dependent Sample T-test. Dari pertanyaan yang telah diproses mendapatkan jawaban dari responden didapatkan jawaban untuk 3 survei tersebut. Uji validasi tersebut yakni sebagai berikut.

Untuk pengujian ini dilakukan survei sebelum dan sesudah menggunakan aplikasi. Hasil dari uji validitas pertanyaan sebelum dan sesudah menggunakan aplikasi masing-masing dapat dilihat pada Tabel 6 dan Tabel 7. Sedangkan untuk hasil uji reliabilitas pertanyaan sebelum dan sesudah menggunakan aplikasi UITEMP masing-masing dapat dilihat pada Tabel 8 dan 9.

TABEL 6

HASIL UJI VALIDITAS PERTANYAAN SEBELUM
MENGUNAKAN APLIKASI UITEMP

	X1	X2	T.Before
Pearson Correlation	1	.300	.783**
Sig. (2-tailed)		.080	.000
N	35	35	35
Pearson Correlation	.300	1	.828**
Sig. (2-tailed)	.080		.000
N	35	35	35
Pearson Correlation	.783**	.828**	1
Sig. (2-tailed)	.000	.000	
N	35	35	35

** . Correlation is significant at the 0.01 level (2-tailed).

TABEL 7

HASIL UJI VALIDITAS PERTANYAAN SESUDAH
MENGUNAKAN APLIKASI UITEMP

Correlations			
	Y1	Y2	T.Aft er
Pearson Correlation	1	.446**	.844**
Sig. (2-tailed)		.007	.000
N	35	35	35
Pearson Correlation	.446**	1	.856**
Sig. (2-tailed)	.007		.000
N	35	35	35
Pearson Correlation	.844**	.856**	1
Sig. (2-tailed)	.000	.000	
N	35	35	35

** . Correlation is significant at the 0.01 level (2-tailed).

Dari hasil uji validitas tersebut didapatkan hasil bahwa pada semua pertanyaan sebelum dan sesudah menggunakan aplikasi adalah valid (nilai sig. < 0.05).

TABEL 8

HASIL UJI RELIABILITAS PERTANYAAN SEBELUM
MENGUNAKAN APLIKASI UITEMP

Case Processing Summary		N	%
Cases	Valid	35	87.5
	Excludeda	5	12.5
	Total	40	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.836	3

TABEL 9

HASIL UJI RELIABILITAS PERTANYAAN SETELAH
MENGUNAKAN APLIKASI UITEMP

Case Processing Summary		N	%
Cases	Valid	35	87.5
	Excludeda	5	12.5
	Total	40	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.866	3

Berdasarkan hasil uji reliabilitas tersebut didapatkan hasil bahwa pada semua pertanyaan sebelum dan sesudah menggunakan aplikasi adalah *reliable* karena nilai Alpha dari sebelum menggunakan aplikasi sebesar 0.836 dan sesudah menggunakan aplikasi sebesar 0.866 sehingga nilai Alpha lebih dari sama dengan 0.78). Sehingga dapat dinyatakan bahwa setiap butir pertanyaan valid untuk disebarkan kepada responden.

Hasil dari survei sebelum dan sesudah menggunakan aplikasi yang berupa perhitungan frekuensi. Kemudian dilakukan perhitungan pengujian menggunakan metode Dependent Sample T-Test melalui aplikasi SPSS untuk mengetahui signifikansi dan reliabilitas dari butir-butir pertanyaan yang diajukan kepada responden dengan ketentuan $H_0: H_{\text{before}} = H_{\text{after}}$ yang berarti sebelum dan sesudah menggunakan aplikasi tidak ada bedanya atau sama dan $H_1: H_{\text{before}} \neq H_{\text{after}}$ yang berarti sebelum dan sesudah penggunaan aplikasi terdapat perbedaan secara nyata. Hasil dari pengujian tersebut dapat dilihat pada Tabel 9.

TABEL 9
HASIL DEPENDENT SAMPLE T-TEST

t-Test: Paired Two Sample for Means		
	Before	After
Mean	31.27941176	12.27941176
Observations	34	34
df	33	
t Stat	26.46051836	
P(T<=t) one-tail	0.00	
t Critical one-tail	1.692360309	

Dari hasil Dependent Sample T-Test tersebut dapat diketahui bahwa Rata-rata atau Mean ketika sebelum menggunakan aplikasi yaitu 31.28 menit lalu setelah menggunakan aplikasi menjadi 12.28 menit, sehingga dapat disimpulkan adanya peningkatan kecepatan yang signifikan dalam mengembangkan aplikasi. Dan jumlah observasi sampel yang digunakan yaitu 35, dengan degree of freedom(df) yaitu 35-1 adalah 34. Hasilnya p value sebesar 0.00. Oleh karena p value lebih kecil dari alpha 5% (0.05) sehingga disimpulkan bahwa terdapat perbedaan yang signifikan waktu antara hasil sebelum dan sesudah menggunakan aplikasi yaitu lebih cepat dengan menggunakan aplikasi ini kecepatan dalam mendesain tampilan ialah lebih cepat dari sebelum adanya aplikasi.

Secara keseluruhan, berdasarkan hasil pengujian manfaat didapatkan beberapa kesimpulan sebagai berikut.

- Penggunaan aplikasi yang digunakan mempercepat membuat desain aplikasi bagi programmer.
- Dengan adanya fitur list template, view, edit dan code dapat mempercepat programmer dalam membuat desain aplikasi ditunjukan pada peningkatan dari sebelum menggunakan aplikasi UTemp ini.
- Pengguna lebih cepat dalam membuat desain aplikasi ditunjukan pada peningkatan saat pengujian membuat simple login baik melalui xml dan code.

- Sebanyak 75% responden menunjukkan tertarik untuk menggunakan aplikasi UTemp ini untuk membuat desain aplikasi.

V. KESIMPULAN

Dari penelitian topik rancang bangun aplikasi *User Interface template* ini didapatkan kesimpulan oleh peneliti adalah sebagai berikut:

- Aplikasi dapat mempercepat pengguna dalam mendesain aplikasinya dengan lima fitur utama yaitu kategori, *view*, *edit*, *code* dan *share*.
- Aplikasi dapat memudahkan pengguna mendapatkan *source code* secara otomatis pada desain aplikasi yang diinginkan

DAFTAR PUSTAKA

- [1] A. Lee and F. Lochovsky, "User Interface Design," *Office Automation*, pp. 3-20, 1985.
- [2] P. Cremonesi, M. Elahi and F. Garzotto, "User interface patterns in recommendation-empowered content intensive multimedia applications," *Multimedia Tools Applications*, vol. 76, no. 4, pp. 5275-5309, 2017.
- [3] A. A. Mahamood, Z. and A. , "Understanding Interface Design Principles and Elements Guidelines: A Content Analysis of Established Scholars," *Proceedings of the Art and Design International Conference (AnDIC 2016)*, 19 June 2018.
- [4] B. R. Suteja and A. Harjoko, "Perancangan User Interface E-Learning berbasis Web," *UPN Veteran Yogyakarta*, pp. 36-46, 28 Mei 2008.
- [5] A. B. Early and M. Zender, "User Interface Design Principles for Interaction Design," *Design Issues*, vol. 24, no. 3, pp. 85-107, 25 June 2008.
- [6] R. Zhou, "User Interface Design Principles for Finnish Websites' Localization in," *Faculty of Nature Sciences*, p. 43, September 2018.
- [7] S. Holla and K. M. Mahima, "Android Based Mobile Application," *International Journal of Computer Trends and Technology*, vol. 31, no. 13, pp. 486-490, 2012.
- [8] R. A. Sagita and H. Sugiarto, "Penerapan Metode Waterfall Pada Sistem Informasi Penjualan Furniture Berbasis Web," *Indonesian Journal on Networking and Security*, vol. 5, no. 4, pp. 49-55, Oktober 2016.
- [9] J. Grossman, "Introduction to Model/View/ViewModel pattern for building WPF apps," 8 October 2005. [Online]. Available: <https://blogs.msdn.microsoft.com/johngrossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/>.
- [10] C. Anderson, "The Model-View-ViewModel (MVVM) Design Pattern," *Pro Business Applications with Silverlight 5*, pp. 461-499, 2012.