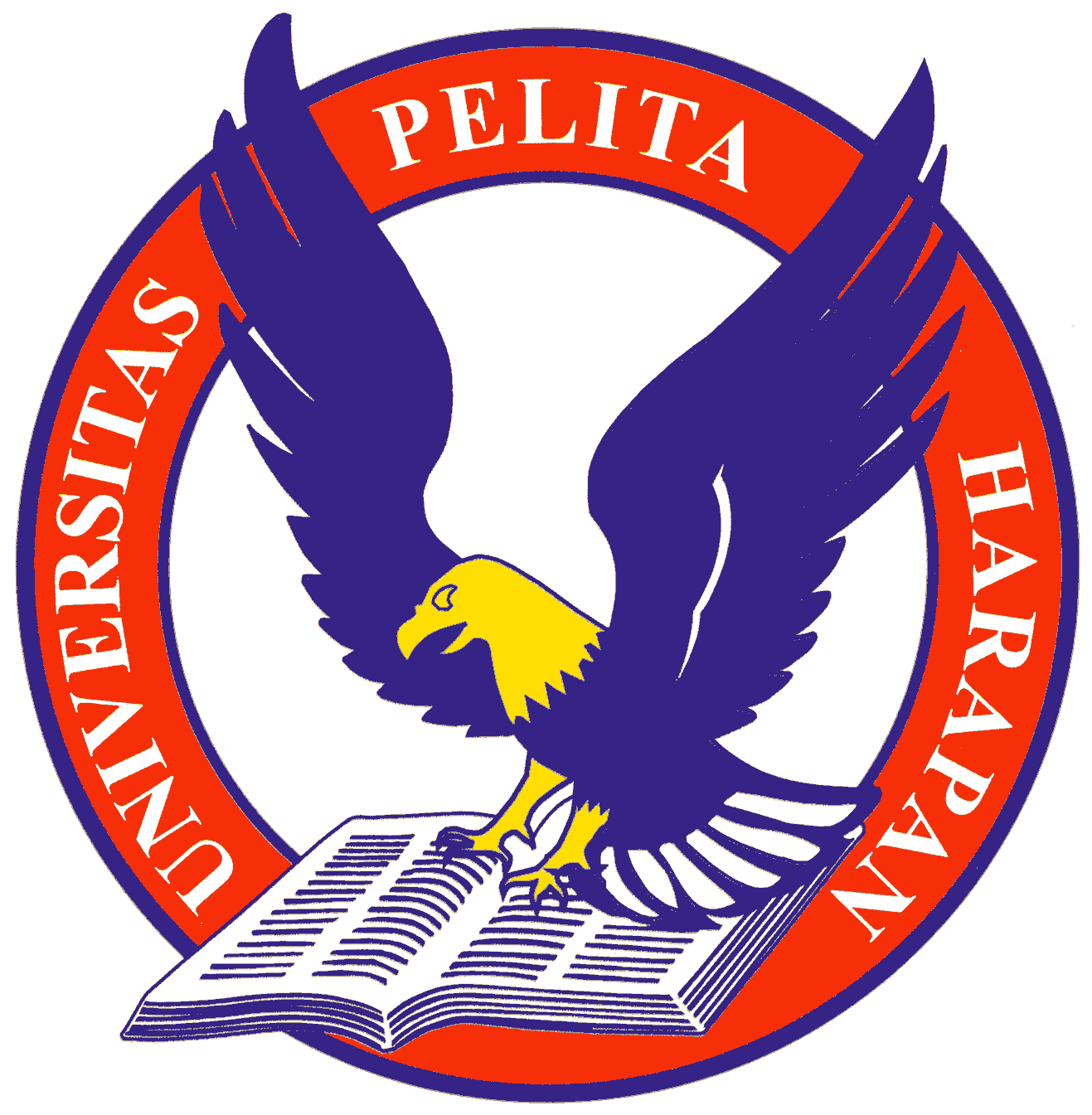
**LAPORAN WHATSCAMP**

**HUMAN COMPUTER INTERACTION – INF 10272**

****

**Disusun oleh:**

ALVIN CHRISTIANTO CALLYSTA – 00000012551

CIKA DESELA PUTRI MARSYAF – 00000011818

CLARISSA VERONICA KUSUMA – 00000013004

JESSICA SEAN – 00000013452

JOSEPH – 00000013230

KEREN ANGELIA RAINTUNG – 00000013847

KEVIN KURNIAWAN – 00000014200

**JURUSAN TEKNIK INFORMATIKA**

**UNIVERSITAS PELITA HARAPAN 2017**

# DAFTAR ISI

[DAFTAR ISI 1](#_Toc500259514)

[DAFTAR FIGURE 3](#_Toc500259515)

[RENCANA DAN PEMBAGIAN KERJA 6](#_Toc500259516)

[RENCANA KERJA 6](#_Toc500259517)

[PEMBAGIAN KERJA 6](#_Toc500259518)

[DESKRIPSI APLIKASI 8](#_Toc500259519)

[BAB I 9](#_Toc500259520)

[LITERATURE SURVEY 9](#_Toc500259521)

[DESIGN USER INTERFACE - Joseph 9](#_Toc500259522)

[HUMAN COMPUTER INTERACTION (HCI) - Alvin Christianto Callysta 14](#_Toc500259523)

[HUBUNGAN HUMAN-COMPUTER INTERACTION (HCI) DENGAN USER INTERFACE (UI) – Clarissa 17](#_Toc500259524)

[UNSUR ESTETIKA DALAM USER INTERFACE (UI) - Kevin Kurniawan 20](#_Toc500259525)

[DATABASE – Jessica Sean 23](#_Toc500259526)

[ASPEK ERGONOMI KOGNITIF PADA HCI - Cika Desela 28](#_Toc500259527)

[SURVEI KELAYAKAN HCI - Keren Angelia Raintung 32](#_Toc500259528)

[BAB II 47](#_Toc500259529)

[FEASIBILITY STUDY 47](#_Toc500259530)

[DATABASE – Jessica Sean 47](#_Toc500259531)

[DESIGN – Clarissa, Joseph, Kevin Kurniawan 51](#_Toc500259532)

[MAIN FEATURE - Cika Desela, Alvin Callysta, Keren Angelia 55](#_Toc500259533)

[BAB III 59](#_Toc500259534)

[DESAIN APLIKASI 59](#_Toc500259535)

[DESAIN AWAL – Kevin Kurniawan 59](#_Toc500259536)

[MODIFIKASI YANG DILAKUKAN (SPLASH SCREEN, MENU, HOME, ABOUT US, DAN LOGIN) – Kevin Kurniawan 63](#_Toc500259537)

[PROTOTIPE – Kevin Kurniawan 74](#_Toc500259538)

[KREATIFITAS DAN INOVASI – Kevin Kurniawan 76](#_Toc500259539)

[REDESIGN – Kevin Kurniawan 78](#_Toc500259540)

[LIMITATION CONSIDERATION – Joseph 82](#_Toc500259541)

[BAB IV 83](#_Toc500259542)

[OTHER WORKING PROGRAM/EXCELLENCE, COMPLEXITY OF CODING – Jessica Sean 83](#_Toc500259543)

[REFERENSI 87](#_Toc500259544)

# DAFTAR FIGURE

[Figure 1. Survey Kelayakan HCI 1. 34](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259545)

[Figure 2. Survey Kelayakan HCI 2. 35](#_Toc500259546)

[Figure 3. Survey Kelayakan HCI 3. 36](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259547)

[Figure 4. Survey Kelayakan HCI 4. 37](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259548)

[Figure 5. Survey Kelayakan HCI 5. 38](#_Toc500259549)

[Figure 6. Perhitungan ANOVA SURVEY 1. 39](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259550)

[Figure 7. Survey Kelayakan HCI 6. 40](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259551)

[Figure 8. Survey Kelayakan HCI 7 40](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259552)

[Figure 9. Survey Kelayakan HCI 8. 42](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259553)

[Figure 10. Survey Kelayakan HCI 9 43](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259554)

[Figure 11. Survey Kelayakan HCI 10. 44](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259555)

[Figure 12. Perhitungan ANOVA SURVEY 2 45](file:///C:\Users\kevin\Downloads\Laporan%20UAS%20-%20WhatsCamp%20FINALLL.docx#_Toc500259556)

[Figure 13. Penyimpanan Data Menggunakan JSON File. [13] 48](#_Toc500259557)

[Figure 14. Contoh User Interface. [28] 52](#_Toc500259558)

[Figure 15. Ilustrasi Fitur yang Dirancang 1. [30] 56](#_Toc500259559)

[Figure 16. Ilustrasi Fitur yang Dirancang 2. [31] 56](#_Toc500259560)

[Figure 17. Ilustrasi Alert Dialog. [36] 58](#_Toc500259561)

[Figure 18. Splash Screen. [37] 59](#_Toc500259562)

[Figure 19. Menu Home. [37] 59](#_Toc500259563)

[Figure 20. Menu Aplikasi (Atas). [37] 60](#_Toc500259564)

[Figure 21. Menu Aplikasi (Bawah). [37] 60](#_Toc500259565)

[Figure 22. Menu Categories. [37] 60](#_Toc500259566)

[Figure 23. Menu Search. [37] 60](#_Toc500259567)

[Figure 24. Halaman Hasil Pencarian. [37] 61](#_Toc500259568)

[Figure 25. Halaman Event. [37] 61](#_Toc500259569)

[Figure 26. Menu About Us. [37] 61](#_Toc500259570)

[Figure 27. Menu Terms and Conditions. [37] 61](#_Toc500259571)

[Figure 28. Menu Settings. [37] 62](#_Toc500259572)

[Figure 29. Menu Login. [37] 62](#_Toc500259573)

[Figure 30. Logo WhatsCamp. [38] 63](#_Toc500259574)

[Figure 31. Folder mipmap. 64](#_Toc500259575)

[Figure 32. Penambahan Image Asset. 64](#_Toc500259576)

[Figure 33. Menghubungkan Image Asset pada layout. 65](#_Toc500259577)

[Figure 34. Splash Screen awal. [37] 65](#_Toc500259578)

[Figure 35. Splash Screen WhatsCamp. 65](#_Toc500259579)

[Figure 36. Item-item dari menu. 66](#_Toc500259580)

[Figure 37. Penambahan Vector Asset. 68](#_Toc500259581)

[Figure 38. Contoh XML dari Vector Asset. 68](#_Toc500259582)

[Figure 39. Contoh menetapkan icon untuk menu pada XML. 69](#_Toc500259583)

[Figure 40. Menu awal. [37] 69](#_Toc500259584)

[Figure 41. Menu modifikasi. 69](#_Toc500259585)

[Figure 42. Component Tree untuk menu Home. 70](#_Toc500259586)

[Figure 43. Isi Home awal. [37] 71](#_Toc500259587)

[Figure 44. Isi Home yang telah dimodifikasi. 71](#_Toc500259588)

[Figure 45. Logo WhatsCamp dengan background. 71](#_Toc500259589)

[Figure 46. Penambahan Image Asset. 72](#_Toc500259590)

[Figure 47. Menambahkan gambar pada layout. 72](#_Toc500259591)

[Figure 48. Isi menu About Us awal. [37] 72](#_Toc500259592)

[Figure 49. Hasil modifikasi menu About Us. 72](#_Toc500259593)

[Figure 50. Tombol Twitter yang dihapus. 73](#_Toc500259594)

[Figure 51. Menu Login awal. [37] 73](#_Toc500259595)

[Figure 52. Hasil modifikasi menu Login. 73](#_Toc500259596)

[Figure 53. Menu Home Eventbrite. [41] 74](#_Toc500259597)

[Figure 54. Fitur search Eventbrite. [41] 74](#_Toc500259598)

[Figure 55. View Event Eventbrite. [41] 75](#_Toc500259599)

[Figure 56. Fitur pembelian tiket Eventbrite. [41] 75](#_Toc500259600)

[Figure 57. Login dengan Facebook atau Google+. 76](#_Toc500259601)

[Figure 58. User dapat menambahkan event. (1) 76](#_Toc500259602)

[Figure 59. User dapat menambahkan event. (2) 76](#_Toc500259603)

[Figure 60. Join Event. 77](#_Toc500259604)

[Figure 61. Telepon contact person. 77](#_Toc500259605)

[Figure 62. activity\_welcome\_screen.xml 78](#_Toc500259606)

[Figure 63. SplashActivity.java 79](#_Toc500259607)

[Figure 64. WelcomeScreenActivity.java 79](#_Toc500259608)

[Figure 65. Intent yang ditambahkan. 80](#_Toc500259609)

[Figure 66. Welcome screen Eventbrite. [41] 81](#_Toc500259610)

[Figure 67. Welcome screen WhatsCamp. 81](#_Toc500259611)

[Figure 68. Login dengan Google+ 82](#_Toc500259612)

# 

# RENCANA DAN PEMBAGIAN KERJA

## RENCANA KERJA

Berikut merupakan rencana kerja yang kelompok kami rancang.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Oct | | | | | Nov | | | | | | Dec | | |
| Week | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | | 2 |
| Desain Interface |  |  |  |  |  |  |  |  |  |  |  | |  |
| Survey |  |  |  |  |  |  |  |  |  |  |  | |  |
| Database |  |  |  |  |  |  |  |  |  |  |  | |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  | |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  | |  |
| Fix Bug dan Error |  |  |  |  |  |  |  |  |  |  |  | |  |
| Dokumentasi |  |  |  |  |  |  |  |  |  |  |  | |  |

## PEMBAGIAN KERJA

Berikut merupakan pembagian kerja dari anggota kelompok kami.

|  |  |  |
| --- | --- | --- |
| Nama | Tugas | |
| Alvin | Literature Survey: Human Computer Interaction (HCI), Feasibility Study: Feature | 2,3,4,5,7,15 |
| Cika | Literature Survey: Aspek Ergonomi Kognitif pada HCI, Feasibility Study: Feature | 3,5,12,8,9 |
| Clarissa | Literature Survey: Hubungan Human Computer Interaction (HCI) dengan User Interface (UI), Feasibility Study: Design Interface | 16,14,11,13 |
| Joseph | Literature Survey: Design User Interface, Feasibility Study: Design Interface | 16,18,23,13,10 |
| Jessica | Literature Survey: Database, Feasibility Study: Database | 20,21,22,16,15 |
| Keren | Literature Survey: Survei Kelayakan HCI, Feasibility Study: Feature | 17,19,1 |
| Kevin K. | Literature Survey: Unsur Estetika dalam User Interface (UI), Feasibility Study: Design Interface | **Coordinator**,  5,25,26,27,6. |

# DESKRIPSI APLIKASI

Nama Aplikasi: **WhatsCamp**

Platform: Android

Deskripsi:

WhatsCamp adalah sebuah program berbasis Android yang berisikan kegiatan-kegiatan yang akan berlangsung pada sebuah kampus (Kampus X). Melalui aplikasi ini, pengguna dapat mengetahui kegiatan-kegiatan yang akan diadakan di dalam kampus dalam jangka waktu tertentu. Pengguna juga dapat melakukan pencarian kegiatan yang berlangsung dan juga dapat memilih apakah pengguna ingin datang dalam kegiatan tersebut.

# BAB I

# LITERATURE SURVEY

## *DESIGN USER INTERFACE* - Joseph

1. **Pendahuluan**

*Interface* dari sebuah system adalah sebuah hal yang penting dalam memenuhi kebutuhan pengguna aplikasi karena menjadi sebuah sistem sehingga sebuah aplikasi dapat digunakan. Menurut [1], tanpa adanya *interface* yang baik dari suatu aplikasi, maka *user* dari program tersebut akan mendapatkan kesulitan untuk mendapatkan atau memberikan informasi kepada sistem sehingga pekerjaan akan menjadi lebih lambat. Oleh karena kebutuhan tersebut harus adanya pertimbangan dalam pembuatan desain dari aplikasi dan juga *interface* yang nantinya akan diimplementasikan kedalam sistem, sehingga pengguna dapat secara maksimal menggunakan aplikasi tersebut.

1. **Prinsip Desain User Interface**

*User Interface* yang baik juga dapat meningkatkan kinerja dari *user* sehingga menjadi efektif dan memberikan *user experience* yang baik juga. Karena hal, tersebut perlu adanya dasar-dasar atau prinsip yang digunakan dalam pembuatan *user interface* pada[1]:

1. *Structure Principle*

*User Interface* harus mempunyai struktur yang baik dalam hal arsitektur model, jelas dan mudah digunakan oleh 1pengguna

1. *Simplicity Principle*

*Outline* dari sebuah *User Interface* haruslah bersifat sederhana, jelas, dapat dimengerti dan juga memudahkan klien dalam menggunakan aplikasi

1. *Visibility Principle*

Konfigurasi dari sebuah *User Interface* memudahkan pengguna untuk tetap dapat berfokus kepada pekerjaan yang dilakukan tanpa terganggu oleh infromasi yang tidak relevan. Desain yang baik dari sebuah *User Interface* adalah desain yang tidak membanjiri pengguna dengan hal-hal yang membuat pengguna menjadi bingung

1. *Feedback Principle*

Setiap hal yang terjadi dalam sistem harus dapat diketahui juga oleh pengguna, seperti apabila terjadi error, ada perpindahan atau perubahan suatu kondisi menggunakan bahasa yang jelas, ringkas dan tidak menimbulkan banyak makna bagi pengguna

1. *Tolerance Principle*

Dengan adanya tingkat toleransi dalam sistem, maka tingkat kesalahan dalam penggunaan aplikasi menjadi berkurang. Setiap tindakan yang pengguna lakukan diperhitungkan dengan baik dan dianalisa secara baik alasan dibalik input dari pengguna

1. *Reuse Principle*

Dalam melakukan desain, maka penggunaan ulang suatu komponan atau sikap Dari *User Interface* harus digunakan. Hal ini memudahkan pengguna dari applikasi untuk menggunakan applikasi secara efektif tanpa harus memikirkan atau mempelajari fitur dari sistem yang beragam.

1. **Konsep Model View Controller**

Dalam *mobile programming,* android merupakan salah satu *operation system* yang cukup sering dipakai sebagai pembelajaran. *Graphical User Interface (GUI)* dari android menerapkan sistem *single-threaded, event-driven* dan dalam *library* yang terbentuk dari kumpulan komponen. Berdasarkan [2], *Android* *User Interface* diatur sedemikian rupa menggunakan *Model-View-Controller Pattern*

1. Model

Sebuah model merepresentasikan sebuah data atau tempat data yang biasa disebut sebagai database dari sebuah sistem.

1. View

View adalah bagian dari sebuah aplikasi yang berfungsi sebagai pihak yang menampilkan tampilan layar, mengirimkan suara ke speaker dan memberikan *feedback* atas tindakan pengguna.

1. Controller

Controller bertugas untuk memberikan respon atas tindakan eksternal seperti *keystroke, screentap,* yang biasa diimplementasikan dalam *event queue.* Controller akan berhubungan dengan controller lain membentuk rangkaian logic yang akan menghubungkan juga dari Model kepada View

1. **User-Centered Design**

*User-Centered Design (UCD)* merupakan sebuah istilah yang menjelaskan sebuah proses desain dimana pengguna atau *end-user* mempengaruhi bagaimana suatu aplikasi akan didesain. Dalam pembuatan sebuah sistem aplikasi terutama dalam hal *User Interface,* pengguna mempunyai peran penting bagaimana aplikasi tersebut nantinya akan berjalan, karena tanpa adanya pengguna maka aplikasi yang sudah dibuat menjadi sia-sia, sehingga pendapat dari pengguna dapat menjadi pertimbangan bagi desainer memutuskan seperti apa aplikasi yang akan dibuat. Menurut Eason (1987) pada [3] ada 3 jenis pengguna :

* Primary : sering menggunakan produk
* Secondary : jarang menggunakan atau lewat orang lain
* Tertiary : dipengaruhi oleh suatu faktor pengenalan

Desainer dari sistem akan mengalami kendala dalam memenuhi seluruh kebutuhan pengguna karena adanya siklus yang terus terjadi dan adanya kebutuhan pengguna yang terus harus dipenuhi sistem. Untuk membantu hal tersebut, maka berdasarkan [3] ada beberapa saran yang dapat digunakan untuk membantu dalam pengumpulan data dengan melibatkan pengguna:

|  |  |  |
| --- | --- | --- |
| **Technique** | **Purpose** | **Stage of the Design Cycle** |
| Background Interviews and questionnaires | Collecting data related to the needs and expectations of users; evaluation of design alternatives, prototypes and the final artifact | At the beginning of the design project |
| Sequence of work interviews and questionnaires | Collecting data related to the sequence of work to be performed with the artifact | Early in the design cycle |
| Focus groups | Include a wide range of stakeholders to discuss issues and requirements | Early in the design cycle |
| On-site observation | Collecting information concerning the environment in which the artifact will be used | Early in the design cycle |
| Role Playing, walkthroughs, and simulations | Evaluation of alternative designs and gaining additional information about user needs and expectations; prototype evaluation | Early and mid-point in the design cycle |
| Usability testing | Collecting quantities data related to measurable usability criteria | Final stage of the design cycle |
| Interviews and questionnaires | Collecting qualitative data related to user satisfaction with the artifact | Final stage of the design cycle |

Kesuksesan dari sebuah desain dapat dikatakan berhasil bila memenuhi mayoritas kebutuhan dari *stakeholder* yang ada, walaupun tidak semua keinginan *stakeholder* dapat terpenuhi.

1. **Kesimpulan**

Berdasarkan sumber yang sudah disurvey, maka dalam pembuatan *User Interface* dari aplikasi memerlukan banyak pertimbangan dan landasan yang selalu berhubungan dengan *end-user* yaitu pengguna. *User Interface* yang dibuat harus didesain sedemikian rupa untuk mempermudah pengguna dalam pemakaian aplikasi serta pertukaran informasi antara pengguna dan aplikasi. Salah satu yang dapat diterapkan dalam pembuatan aplikasi kami adalah 6 prinsip yang digunakan untuk pembuatan desain *User Interface*. Untuk penerapan *User-Centered Desain* tidak akan diterapkan secara penuh karena keterbatasan waktu yang ada dalam pembuatan program kami, akan tetapi interaksi dan pengumpulan informasi dengan pengguna untuk pembuatan dan desain dari *User Interface* aplikasi akan kami usahakan untuk tetap terjadi.

## *HUMAN COMPUTER INTERACTION (HCI)* - Alvin Christianto Callysta

**1.** **Pendahuluan**

Human-computer interaction (HCI) [42] adalah sebuah studi dalam ilmu komputer yang mencakup tentang berbagai hal yang berhubungan dengan ilmu kognitif (pikiran, respon, penilaian) dan faktor manusia di dalam interaksi di antara manusia dengan komputer. HCI umumnya fokus kepada konsep tentang *usability*, yang berarti bagaimana sebuah produk dapat membuat manusia dapat mencapai hal-hal tertentu dengan lebih mudah, dalam hal ini berarti bagaimana cara meningkatkan usabilitas *software* aplikasi yang akan dibuat*.* Sebagai pengembang aplikasi*,* diperlukan pemeriksaan terhadap aplikasi yang sedang dikembangkan. Apakah fitur dari aplikasitsb. telah memenuhi standar usabilitas yang telah dibuat, serta bisa bekerja secara efektif dan membantu *user* mencapai tujuannya dengan mudah.

1. **Definisi Usabilitas**

Ada beberapa definisi dari usability, berdasarkan ISO (*International Organization for Standarization)* diantaranya [43]:

1. ISO 9241-11: GUIDANCE ON USABILITY (1998)

Usability: bagaimana sebuah produk dapat digunakan oleh beberapa user untuk mencapai goal spesifiik secara efektif, efisien, dan kepuasan dalam konteks penggunaannya.

ISO 9241-11 menjelaskan bagaimana mengidentifikasi informasi yang perlu diambil ketika sedang mengevaluasi usabilitas dalam hal pengukuran performa dan kepuasan user.

2. ISO/IEC 9126: SOFTWARE PRODUCT EVALUATION\*QUALITY CHARACTERISTICS AND GUIDELINES FOR THEIR USE (1991)

ISO/IEC 9126, berkembang secara terpisah sebagai sebuah standar dari *software engineering,* mendefinisikan usabilitas sebagai suatu kontribusi independen untuk kualitas software yang dikaitkan dengan desain dan evaluasi dari interaksi dan *user interface*:

Usability: suatu kumpulan atribut penunjang upaya yang dibutuhkan untuk penggunaan, dan untuk penilaian individual, yang dinyatakan atau diimplikasikan oleh beberapa user.

Namun karena usabilitas terlalu abstrak untuk dipelajari secara langsung, biasanya usabilitas terbagi menjadi beberapa atribut seperti yang telah disebutkan di atas:

a. *Learnability*: Seberapa mudah untuk mempelajari fungsi dari fitur dan menjadi mahir dalam menggunakan fitur untuk menyelesaikan pekerjaan.

b. *Efficiency*: Jumlah pekerjaan per satuan waktu yang bisa dikerjakan user ketika menggunakan fitur tersebut.

c. *User retention over time*: Seberapa baik user mengingat cara penggunaan fitur setelah beberapa periode tidak menggunakannya.

d. *Error rate*: Seberapa rendah jumlah error yang dilakukan user ketika melakukan pekerjaan.

e. *Satisfaction*: Menunjukkan kesan subjektif dari user mengenai fitur yang telah dibuat. [44]

**3.** **Penerapan konsep Usabilitas terhadap Fitur Aplikasi**

Fitur yang baik adalah fitur yang memiliki usabilitas yang cukup baik, sehingga user bisa memanfaatkan aplikasi dalam menyelesaikan pekerjaan secara maksimal. Oleh karena itu, ada beberapa metode yang bisa digunakan untuk menjelaskan usabilitas dari sebuah fitur dan apakah fitur tersebut bekerja dengan baik dan memenuhi standar atribut usabilitas, di antaranya [45]:

a. *Heuristic evaluation*: adalah metode yang paling informal dan melibatkan juri atau pengamat ahli, mempertanyakan apakah setiap elemen dialog mengikuti prinsip dari usabilitas.

b. *Cognitive walkthroughs*: menggunakan prosedur detail yang lebih eksplisit untuk mensimulasikan proses *problem solving* dari user di setiap dialog, memeriksa jikalau goal dari user yang telah disimulasikan bisa diasumsikan agar menuju langkah selanjutnya dengan benar.

c. *Formal usability inspections*: menggunakan enam langkah prosedur dengan peran yang telah dibuat secara ketat untuk menggabungkan *heuristic evaluation* dan versi simpel dari *cognitive walkthroughs*.

d. *Pluralistic walkthroughs*: adalah pertemuan di mana user, *developer*, dan orang-orang *human factor* menjalani skenario, membahas tiap elemen dialog.

e. *Feature inspection*: daftar dari fitur yang digunakan untuk mencapai tugas terntentu, pengecekan langkah yang seharusnya tidak mungkin dilakukan user, dan langkah yang membutuhkan pengetahuan atau pengalaman yang lebih untuk menilai fitur yang akan dibuat.

f. *Consistency inspection*: perancang desain merepresentasikan proyek-proyek untuk menginspeksi sebuah interface untuk melihat apakah berjalan sesuai yang telah dirancang.

g. *Standards inspection*: melakukan inspeksi dengan melibatkan ahli yang mengetahui standar interface atau fitur menginspeksi produk dan melihat apakah produk tersebut memenuhi kriteria.

1. **Kesimpulan**

Berdasarkan sumber yang sudah disurvey, maka penerapan HCI di dalam pembuatan *feature* dari aplikasi, kami akan menggunakan metode *feature inspection* untuk memastikan apakah fitur yang telah kami buat telah memenuhi atribut dasar dari usabilitas yang telah disebutkan, agar supaya kami mengetahui seberapa maksimal kemampuan fitur dalam membantu user mencapai goalnya. Fitur yang baik setidaknya memiliki satu diantara lima atribut yang diantaranya adalah *learnability, efficiency, user retention over time, error rate, satisfaction*.

## HUBUNGAN *HUMAN-COMPUTER INTERACTION (HCI)* DENGAN *USER INTERFACE (UI)* – Clarissa

1. **Pendahuluan**

*Human-computer interaction (HCI)* merupakan sebuah konsep penting dalam pembangunan *User Interface*, yang merupakan “wajah” dari sebuah aplikasi. HCI memadukan konsep dan metode dari *computer science* dan *design* untuk membangun sebuah *interface* yang mudah digunakan, efisien, dan mudah diakses. Dengan menerapkan prinsip HCI dalam pembuatan *interface*, kita dapat menghasilkan sebuah interface yang menarik, mudah dimengerti, dan efisien.

1. **Hubungan Prinsip *Human-Computer Interaction* (HCI) dengan *User Interface* (UI)**

Salah satu alasan mengapa sebuah aplikasi dapat memiliki *interface* yang kurang baik adalah karena orang – orang yang mengembangkan *interface* aplikasi kurang melakukan latihan yang didasarkan oleh HCI [4]. Dari hal ini, dapat dikatakan bahwa HCI memiliki peran penting dalam membangun sebuah *interface*. Untuk merancang dan mengimplementasikan sebuah *interface* yang efektif, ada beberapa prinsip HCI yang penting dan perlu diterapkan :

1. Mental Models

*Designer* perlu mengingat pengetahuan yang dimiliki oleh pengguna dari pengalamannya dalam dunia nyata dan bagaimana hal tersebut dapat diaplikasikan pada *interface.* Praktisi HCI konsentrasi pada definisi mental model sebagai sebuah set kepercayaan bagaimana sistem bekerja. Pengguna berinteraksi dengan sistem berdasarkan kepercayaan tersebut.

1. Metaphors

Beberapa karakteristik dari mental model pengguna adalah metafora. Metafora atau kiasan menyediakan jalan pintas dalam memahami konsep sulit, dan hal ini dapat digunakan untuk membentuk perilaku pengguna dalam situasi yang mereka tidak kenal.

1. Visibility

Berdasarkan prinsip dari *visibility, user interface* harus selalu membantu pengguna mengerti kondisi dari sistem saat itu dan operasi yang dapat dilakukan. Sebagai contoh, jika pengguna mengarahkan cursor ke salah satu bagian layar, seharusnya pengguna sudah mengetahui dengan jelas apa yang akan terjadi jika ia mengklik mouse.

1. Affordance

*Perceived affordance* adalah kualitas yang memudahkan pengguna untuk melihat dan mengenali fungsionalitas yang ditawarkan *interface*.

1. Feedback

*Feedback* adalah informasi mengenai respons sistem dari segala aksi yang dilakukan yang seharusnya diberikan kepada pengguna [5].

1. **GOMS (*Goals, Operators, Methods, and Selection*)**

GOMS (Goals, Operators, Methods, and Selection) merupakan salah satu model teoretis yang paling populer dalam lapangan HCI. GOMS digunakan untuk meningkatkan efisiensi dari interaksi manusia-mesin dengan mengidentifikasi dan menghilangkan aksi pengguna yang tidak diperlukan. Model GOMS ini meliputi elemen-elemen sebagai berikut :

1. Goal

Sebuah spesifikasi dari apa yang pengguna ingin capai, seperti mencari sebuah kata dari sebuah dokumen.

1. Operators

*Software* memperbolehkan aksi-aksi yang dapat diambil atau dilakukan pengguna, seperti *input keyboard.*

1. Method

Metode yang dapat digunakan untuk menyelesaikan sebuah tujuan. Sebagai contoh, untuk menemukan sebuah kata dalam sebuah dokumen, pengguna dapat menggunakan fitur *Find* dalam MS Word, atau pengguna dapat melakukan *scroll down* dan mencari kata tersebut secara manual.

1. Selection

Peraturan – peraturan yang harus dituruti oleh pengguna dalam memilih metode mana yang akan digunakan dalam situasi tertentu. Sebagai contoh, pengguna dapat memilih antara 2 pilihan ini :

1. Jika dokumen tersebut panjang, maka gunakanlah fitur *Find.*

B. Jika dokumen tersebut pendek, maka gunakanlah *function scroll*, karena hal tersebut lebih cepat. [6]

1. **Kesimpulan**

Berdasarkan *literature survey* mengenai hubungan antara HCI dengan *User Interface*, kami menemukan bahwa penerapan prinsip HCI sangat penting dalam perancangan *interface*. Dengan diterapkannya prinsip HCI dalam perancangan *interface* aplikasi kami, kami dapat membuat sebuah aplikasi yang mudah dimengerti dan efisien. Selain itu, kami juga menemukan bahwa hal terpenting dalam pembentukan atau perancangan sebuah *interface* adalah mudah dimengerti. *Interface* yang baik bukan dilihat dari seberapa indah atau cantik tampilannya, melainkan apakah pengguna dapat langsung mengerti fungsi dari setiap fitur yang telah disediakan.

## UNSUR ESTETIKA DALAM *USER INTERFACE* (UI) - Kevin Kurniawan

1. **Pendahuluan**

Kita dapat melihat bahwa terdapat beberapa aplikasi yang ada di pasaran menonjolkan unsur estetika yang memberikan kenyamanan dan pengalaman tersendiri bagi penggunanya. Unsur estetika ini merupakan salah satu hal esensial untuk dipertimbangkan dalam melakukan perancangan *user interface* dari aplikasi. Unsur estetika ini juga mempengaruhi minat pengguna terhadap penggunaan aplikasi tersebut.  
 Ensiklopedia Britannica menjelaskan estetika dalam konteks seni, sebagai “pembelajaran filosofi tentang keindahan dan rasa”. Definisinya juga menjelaskan bahwa agar estetika itu ada, maka harus ada obyek, penerima, dan sebuah pengalaman. Maka dari itu, obyek estetik, dengan caranya, mengkomunikasikan kepada penerimanya dalam bentuk emosi. Jika *interface* adalah obyek estetik dan pengguna merupakan penerima estetik, maka harus ada interaksi di antaranya untuk memunculkan pengalaman. [7]

1. **Pentingnya Unsur Estetika dalam *User Interface* (UI)**

Unsur estetika dapat mendukung kepopuleran dan penerimaan masyarakat terhadap penggunaan suatu aplikasi. Unsur estetika diperlukan dalam *user interface* suatu aplikasi bukan tanpa alasan. Berdasarkan [8], berikut merupakan beberapa alasan mengapa unsur estetika penting untuk diterapkan dalam *user interface* suatu aplikasi.

* Estetik yang baik membuat pengguna tertarik dan dengan demikian membuat GUI (*Graphical User Interface*) lebih popular.
* Gambar pada GUI (*Graphical User Interface*) membantu mengidentifikasi dan mengingat elemen GUI yang berbeda.
* Jika unsur estetika-nya cocok dengan yang pengguna inginkan dan personalitasnya, maka pengguna akan menikmati yang dilakukannya sambil mengoperasikan GUI.
* Warna yang menenangkan mengurangi ketegangan mata dan membantu pengguna untuk bekerja dalam waktu yang lebih lama.
* Berkontribusi dalam aspek lainnya, seperti kemudahan dalam penggunaan, efisiensi, dan efektifitas dari GUI. [8]

1. **Fitur Ideal dari Unsur Estetika pada *User Interface* (UI)**

Unsur estetika dalam *user interface* dapat dimunculkan melalui komponen-komponen di dalam aplikasi. Berdasarkan [8], berikut merupakan beberapa fitur ideal yang sebaiknya ada untuk meningkatkan unsur estetika dalam *user interface*.

* GUI (*Graphical User Interface*) harus indah dan menarik bagi semua pengguna dalam setiap waktu.
* Tampilan dan rasa dari GUI harus sesuai dengan tampilan dan rasa dari GUI sistem operasi.
* Tampilan dan rasa dari GUI tidak boleh bertentangan dengan tampilan dan rasa dari GUI lainnya yang berjalan pada komputer.
* Tampilan dan rasa dari GUI sebaiknya sesuai dengan personalitas (atau pilihan pribadi, perasaan, dan emosi) dari pengguna.
* Idealnya, tampilan dan rasa dari GUI menghasilkan efek yang menenangkan sehingga menyalurkan energi bagi pengguna untuk bekerja dalam waktu yang lama.
* GUI sebaiknya menggunakan *icon* gambar dan *background* gambar agar lebih menarik.
* Gambar yang digunakan pada elemen GUI sebaiknya menyerupai dengan obyek aslinya untuk membantu pemahaman yang lebih mudah, sebagai contoh, *printer icon* terlihat seperti *printer*, *trash icon* terlihat seperti kotak sampah, dan sebagainya.
* Elemen GUI sebaiknya berbentuk tiga dimensi dan memiliki *background* animasi untuk memberikan rasa yang realistis. [8]

1. **Metode Peningkatan Unsur Estetika pada *User Interface* (UI)**

Selain memberikan fitur ideal dalam aplikasi, sebagaimana yang telah dibahas pada bagian sebelumnya, pengembang aplikasi juga dapat melakukan beberapa aksi untuk untuk meningkatkan unsur estetika dalam *user interface*. Berdasarkan [8], berikut merupakan aksi-aksi yang berupa metode yang dapat diterapkan.

* Menggunakan warna yang tepat pada tempat yang tepat.
* Menggunakan *background* gambar dan *icon* gambar.
* Menggunakan 3D *desktop*, 3D *windows*, dan 3D *icons*.
* Menggunakan animasi, seperti animasi pada *icon*, animasi pada *desktop* (seperti *virtual reality*), animasi pada komponen-komponen (seperti *sliding out toolbar*).
* Menambahkan audio stereoponis, suara, dan mekanisme tambahan lainnya.
* Meningkatkan *layout* dari *windows* dan *icon* pada *desktop*.
* Meningkatkan *layout* dari tombol pada *toolbar* dan *function palettes*.
* Menyusun elemen yang berhubungan pada kelompok seperti dalam *toolbar*, *function palettes*, dan lain sebagainya. [8]

1. **Kesimpulan**

Berdasarkan beberapa bagian dari survei yang telah dibahas pada bagian sebelumnya, kami menyadari pentingnya penerapan unsur estetika dalam *user interface* dari aplikasi yang akan kami buat. Beberapa alasan di atas mengenai pentingnya unsur estetika, membuat kami menyimpulkan bahwa unsur estetika ini diperlukan untuk memberikan daya tarik bagi pengguna terhadap aplikasi dan membantu pengguna agar dapat lebih efektif dalam menggunakan aplikasi. Perancangan dan penerapan unsur estetika untuk *user interface* ini tentunya harus tetap sesuai dan saling mendukung terhadap prinsip *Human Computer Interaction* (HCI) yang dibahas pada bagian *Literature Survey* mengenai *Human Computer Interaction* sendiri.

## DATABASE – Jessica Sean

* + - 1. Pendahuluan

Database dari sebuah sistem merupakan hal yang penting dalam sebuah aplikasi yang memiliki tujuan utama untuk menyimpan segala data yang dibutuhkan dalam mengoperasi atau menjalankan sebuah aplikasi. Database merupakan struktur komputer yang dapat dipakai bersama-sama *(shared)* yang menyimpan:

* Data *end-user*
* Metadata, atau yang disebut sebagai data mengenai data, dimana data end-user diintegrasi dan diatur. Metadata tersebut mendeskripsikan karakteristik dan hubungan (*relationship*) yang menghubungkan data yang ada pada database. [9]
  + - 1. Database Management System (DBMS)

*Database Management System* merupakan suatu sistem *software* yang memperbolehkan pengguna untuk mendefinisikan, membuat, me-*maintain* dan mengontrol akses ke database [10]. Berikut merupakan sebuah persamaan yang dapat meilustrasikan mengenai DBMS:

Database + DBMS software = Database system [10]

Terdapat berbagai keuntungan yang diperoleh dengan menggunakan DBMS, yaitu:

* Data *sharing* yang lebih baik. DBMS membantu untuk membuat sebuah lingkungan dimana *end user* dapat memiliki akses yang lebih baik dan data yang lebih terstruktur.
* Data *security* yang lebih baik. DBMS menyediakan *framework* untuk privasi data dan ketentuan keamanan yang lebih baik.
* Mengurangi inkonsistensi data. Inkonsistensi data terjadi ketika terdapat data dengan berbagai versi di berbagai tempat yang berbeda. Dengan database yang didesain dengan benar, kemungkinan terjadinya inkonsistensi data dapat dikurangi.
* Meningkatkan akses data. Dengan menggunakan DBMS, *end user* dapat dengan mudah menerima jawaban dari *ad hoc query*, sehingga memudahkan *end user* untuk mencari data tertentu.
* *Decision making* yang lebih baik. DBMS menyediakan framework yang memfasilitasi inisiatif kualitas data, sehingga *end user* dapat mengakses data yang telah di-*improve* dan diatur dengan baik untuk mendapatkan informasi dengan kualitas yang baik. Dengan demikian, informasi tersebut dapat mempengaruhi *decision making*.
* Meningkatkan produktivitas *end user*. Tersedianya data dan *tools* yang dapat mentransformasi data menjadi informasi yang digunakan dapat mempermudah *end user* dalam membuat keputusan yang lebih cepat. [9]

Dalam perancangan sebuah sistem, memilih DBMS yang tepat merupakan hal yang sangat penting. Tahap pemilihan DBMS harus dimulai dengan mencari kebutuhan sistem dari segi database. Dengan mengetahui kebutuhan dari sistem, perancang database dapat melihat fitur apa saja yang dibutuhkan oleh sistem dari DBMS dan memilih DBMS yang sesuai. Langkah yang paling penting dalam memilih DBMS adalah dengan menggunakan DBMS sistem yang akan memberikan keuntungan bagi sistem [11].

* + - 1. MySQL Database

MySQL merupakan database relasional yang banyak digunakan dalam aplikasi, terutama aplikasi berbasis web dengan PHP. MySQL merupakan pilihan database yang populer karena kelebihan dari MySQL yang dijabarkan sebagai berikut [12]:

* Tidak dikenakan biaya, sehingga semua orang dapat men-*download* MySQL dari http://www.mysql.com tanpa biaya.
* Cepat, *robust* dan *scaleable*. Walaupun kecepatan MySQL dioptimasi, MySQL tetap dapat menyimpan ratusan sampai ribuan tabel yang menyimpan jutaan data *entry*.
* Instalasi mudah dilakukan dengan *executeable binaries* yang tersedia dalam berbagai platform OS, termasuk Windows, Linux, Solaris, Mac OS, dan lain sebaginya.
* *High Portability*. Database MySQL mudah dipindahkan dari satu mesin ke mesin lainnya tanpa mengurangi performa.

Namun, database yang kurang didesain dengan matang dapat berakibat kepada penurunan performa dari database MySQL. Oleh karena itu, dibutuhkan desain database yang baik, termasuk cara database melakukan operasi dalam mengambil data [12].

* + - 1. SQLite Database

SQLite merupakan DBMS yang banyak digunakan pada berbagai platform karena naturnya yang ringan (*lightweight*). Banyak platform yang mendukung SQLite, diantaranya adalah Android OS [13]. SQLite merupakan DBMS yang digunakan sebagai sebuah aplikasi format file, database untuk *gadget* dan *website*, dan RDBMS (*Relationship Database Management System*). SQLite menyediakan berbagai fitur, di antaranya adalah:

* Tidak perlu adanya konfigurasi. SQLite di desain sedemikian rupa sehingga DBMS tersebut tidak membutuhkan file konfigurasi. SQLite juga tidak membutuhkan adanya tahap instalasi ataupun setup awal. SQLite tidak menggunakan server dan langsung terlekat di dalam aplikasi, sehingga tidak ada administrator yang diperlukan untuk me-*maintain* sebuah instansi database ataupun *permission* untuk pengguna.
* Tidak adanya copyright. Source code dari SQLite merupakan domain yang publik, sehingga siapapun dapat memodifikasi, mendistribusi ataupun menjual code.
* *Cross-platform*. Fitur ini merupakan fitur yang membuat SQLite banyak digunakan oleh para developer. Dengan menggunakan SQLite, file database dapat dipindahkan ke dalam sistem lainnya yang berjalan di atas arsitektur yang berbeda. Hal ini memungkinkan karena *database file format* dari SQLite merupakan binary dan semua mesin menggunakan format yang sama.
* *Compact*. SQLite database merupakan *disk file* yang umum. SQLite database dapat dijalankan tanpa adanya server dan di design untuk menjadi ringan dan simple. Hal ini dapat dilihat melalui SQLite versi 3.7.8 yang memiliki *footprint* kurang dari 350KiB (kibibyte).
* *Fool proof*. Basis code SQLite diberikan *comment* dengan baik, mudah dimengerti serta *modular*. [13]
  + - 1. Properti ACID pada DBMS

Transaksi merupakan sebuah unit dari program yang melakukan *update* terhadap beragam item data [14]. Untuk memastikan integritas data pada transaksi, sistem database harus menjaga beberapa properti berikut [15]:

* + - 1. *Atomicity*

Properti ini memegang prinsip bahwa sebuah transaksi harus terjadi secara keseluruhan atau tidak sama sekali, sehingga tidak ada transaksi yang hanya terjadi sebagian [14]. Hal ini dikarenakan transaksi dianggap sebagai satu unit yang dijalankan secara lengkap atau tidak dijalankan sama sekali. Ilustrasi yang dapat menjadi contoh adalah sistem bank, dimana akun A memiliki saldo sejumlah $400 dan B memiliki $700 dan A akan melakukan transfer kepada B sebesar $100. Terdapat dua operasi yang harus dijalankan dalam kasus ini, yaitu untuk mengurangi saldo A dan menambahkan ke saldo B. *Atomicity* memiliki arti bahwa dua operasi ini harus keduanya dilakukan secara berhasil atau transaksi akan gagal (tidak mengeksekusi operasi sama sekali) [15].

Properti ini melibatkan dua operasi, yaitu [14]:

* *Abort*: ketika transaksi di-*abort*, maka perubahan pada database tidak terlihat.
* *Commit*: jika transaksi berhasil *commit,* maka perubahan yang dibuat pada database dapat terlihat.
  + - 1. *Consistency*

Properti ini mengandung arti bahwa integritas dari batasan harus dijaga dengan baik sehingga database bersifat konsisten sebelum dan sesudah transaksi. Untuk menjaga konsistensi dari database, eksekusi dari transaksi harus dijalankan dalam isolasi, dimana tidak ada transaksi lain yang jalan bersamaan ketika satu transaksi sedang berjalan [15].

* + - 1. *Isolation*

Untuk setiap pasang transaksi, satu transaksi harus berjalan setelah transaksi sebelumnya diselesaikan. Hal ini untuk mendukung adanya konsistensi dari database [15].

* + - 1. *Durability*

Properti ini memastikan bahwa ketika sebuah transaksi sudah dieksekusi secara penuh, *update* dan modifikasi pada database akan disimpan dan di-*write* ke disk, sehingga perubahan pada database bersifat permanen walaupun ada terjadi *system crash* atau *system failure* [14].

1. Kesimpulan

Sebagai kesimpulan, database merupakan unsur yang sangat penting yang harus didesain dengan baik dalam perancangan dari sebuah sistem. Hal ini dikarenakan kesalahan pada desain dan penerapan database dapat memberikan pengaruh yang cukup besar kepada aplikasi yang sedang dibuat. Dalam memutuskan DBMS yang akan digunakan pada sebuah sistem, diperlukan juga pertimbangan terhadap fitur-fitur yang dibutuhkan oleh sistem/aplikasi yang akan dibuat, sehingga DBMS yang dipilih dapat memebuhi semua kebutuhan sistem dari segi database. Selain itu, database dari sebuah sistem juga dipastikan harus dilengkapi oleh properti ACID, agar semua transaksi yang terdapat di dalam database dapat berjalan dengan baik dan sistem dapat berjalan dengan baik dan sesuai dengan fungsinya.

## ASPEK ERGONOMI KOGNITIF PADA HCI - Cika Desela

1. **Pendahuluan**

HCI mencari dukungan antara manusia dan komputer dengan cara membuat komputer bekerja secara efektif untuk melaksanakan pekerjaannya. Salah satu masalah yang menjadi perhatian antara pengguna dengan komputer adalah pada sistem interaktifnya. Masalah kebiasaan pengguna dan efektivitas & produktivitas sistem menjadi perhatian utama yang mendasari pengembangan interaksi manusia dengan komputer. Untuk mengatasi permasalah tersebut, maka HCI telah menyediakan subjek dan serangkaian masalah yang merupakan stimulus penting bagi pengembangan ergonomi kognitif pada HCI. Berikut adalah penjelasan mengenai aspek ergonomi kognitif dan kontribusinya sebagai pengembangan bagi HCI.

1. **Hubungan HCI dengan Ilmu Pengetahuan Kognitif**

Menurut [16] HCI dan ilmu pengetahuan kognitif memiliki kesamaan dalam faktor manusia. HCI berfokus pada hal-hal terapan sedangkan ilmu kognitif berfokus pada masalah teoritis. Pengguna aplikasi memahami informasi, menggunakan pengetahuan, dan membuat keputusan yang semuanya merupakan domain dari ilmu kognitif. Pemahaman pengguna aplikasi memungkinkan pengembang aplikasi untuk memprediksi dan menjelaskan interaksi yang terjadi antara pengguna dan sistem komputer. Tanpa pemahaman itu, sulit untuk memfasilitasi interaksi yang optimal dalam perancangan aplikasi.

1. **Area interferensi Aspek Ergonomi Kognitif pada HCI**

Aspek ergonomi kognitif yang berkontribusi penting bagi HCI adalah untuk mendukung aspek dimana interaksi bergantung pada pengetahuan yang diperlukan bagi manusia untuk menggunakan komputer agar dapat berjalan dengan efektif. Berdasarkan [17] ada 3 area interferensi pada aspek kognitif HCI;

1. *Design*

Dari sudut pandang ergonomi kognitif, ada dua aspek yang menarik dalam perancangan sistem. Pertama proses dari desain, aspek ergonomi kognitif ingin memahami bagaimana orang merancang sebuah sistem baru serta faktor individu apa dan kelompok apa yang terlibat dalam membuat keputusan yang mengarah pada solusi tertentu yang menentukan sistem. Kedua, Para ahli ergonomi kognitif ingin mengetahui apakah solusi yang diadopsi sesuai dengan kebutuhan dan karakteristik pengguna. Peran mereka dalam pengertian ini adalah untuk menggambarkan manusia di semua tingkat organisasi fungsional yang sesuai dengan sistem yang dirancang [18] [19].

1. *Technological Innovation*

Konsep “*User-centered design”* yang muncul pada tahun 1980-an bertujuan untuk menggambarkan manusia yang berinteraksi dengan sistem dari sudut pandang pengetahuan kognitif. Namun, perkembangan teknologi memaksa ergonomi kognitif untuk memikirkan kembali konsep interaksi seperti;

* Aplikasi didesain untuk digunakan dalam lingkungan yang cerdas dan *mobile.*
* Aplikasi dapat didasarkan pada spektrum material baru yang luas, seperti *Transparant Organic Light Emitting Diodes* (TOLED).
* Aplikasi perlu dapat untuk beradaptasi dengan intensi, keadaan pengetahuan dan emosi pengguna saat ini.

1. *Safety and Accident Investigation*

Ada 3 macam pendekatan untuk mempelajari *human error*:

1. *Human Reliability Analysis* (HRA)

Teknik ini berdasarkan pada asumsi bahwa tindakan dari seseorang di tempat kerja dapat dipertimbangkan dari sudut pandang yang sama seperti operasi mesin. Tujuan dari pendekatan ini adalah untuk memprediksi kemungkinan kesalahan manusia dan mengevaluasi bagaimana keseluruhan sistem kerja terdegradasi akibat kesalahan ini sendiri atau dalam kaitannya dengan pengoperasian mesin, karakteristik tugas, perancangan dan karakteristik sistem individu [20].

1. Pendekatan yang diadaptasi dari kognitif psikologi. Sudut pandang ini menyarankan untuk menyelidiki penyebab kesalahan manusia dengan menganalisis karakteristik pemrosesan informasi manusia. Menurut [21], ada 3 jenis kesalahan tergantung pada tingkat dan tingkat kendali kognitif yang terlibat dalam perilaku yang keliru:
2. *Errors based on skills*: error ini terjadi apabila tingkat kesadaran seseorang menjadi rendah karena terbiasa dengan sebuah pekerjaan.
3. *Errors based on rules:* aktivasi aturan yang benar tergantung pada interpretasi kondisi yang situasional. Jika situasi disalahartikan, aturan yang diambil juga tidak sesuai. Dengan kata lain, kesalahan berdasarkan aturan akan terjadi.
4. *Errors based on knowledge*: error ini terjadi apabila sesorang menghadapi situasi masalah yang baru sehingga apabila tindakan untuk mencari resolusinya tidak direncanakan dengan benar, kesalahan berbasis pengetahuan akan terjadi.
5. Menurut pendekatan ini, manusia dan lingkungan kerjanya harus dianggap sebagai sistem kognitif gabungan yang interaktif [22]. Berdasarkan asumsi ini, beberapa ahli mengusulkan metodologi untuk memperkirakan probabilitas kesalahan manusia tergantung pada situasi spesifik dimana interaksi antar manusia terjadi. Metodologi mengandaikan dua tahap analisis: (1) mengidentifikasi jenis kesalahan pekerjaan yang mungkin terjadi untuk tugas tertentu dalam sebuah skenario kejadian. (2) mengklasifikasikan probabilitiasnya untuk mengidentifikasi mana yang paling mungkin dan mana yang paling kecil kemungkinannya dalam sistem kognitif gabungan [23].
6. **Kesimpulan**

Berdasarkan *literature study* mengenai aspek ergonomi kognitif pada HCI, kelompok kami akan mencoba untuk mengimplementasi pendekatan pada area interferensi *Safety and Accident Investigation* yang mempelajari kemungkinan *human error* bagi pengguna aplikasi *WhatsCamp* agar kami dapat memprediksi kesalahan yang disebabkan oleh pengguna aplikasi dan solusi untuk mengatasinya. Kelompok kami akan menggunakan prinsip aspek ergonomi dan kognitif pada *project* HCI dengan alasan pertimbangan bahwa permasalahan mempelajari kebiasaan pengguna mengenai efektivitas dan produktivitas sistem sangatlah berpengaruh pada penyelesaian *project* kami dan aspek ergonomi kognitif menyediakan solusi dan pendekatan untuk menyelesaikan masalah tersebut.

## SURVEI KELAYAKAN HCI - Keren Angelia Raintung

1. **Pendahuluan**

Survei sangat dibutuhkan dalam segala aspek ketika melakukan sebuah proyek. Sebuah survei juga dapat menentukan kesuksesan dari aplikasi tersebut. Pada [42] dapat diketahui bahwa konsep survei merupakan strategi efektif untuk mengukur konsepsi alternatif dengan cepat sehingga bisa mendapatkan umpan balik yang didapat, sekaligus memberikan informasi kepada kelompok kami untuk menyesuaikan fitur-fitur pada aplikasi apakah aplikasi kami berjalan dengan baik. Tapi, yang terpenting, hasil survei ini bisa menjadi tolak ukur dalam pengembangan aplikasi.

1. **Tujuan dan Manfaat Dilakukannya Survei**

Survei dilakukan dengan tujuan untuk mengambil informasi yang bermanfaat untuk kemajuan fitur dari aplikasi yang dijalankan. Pada [43] di jelaskan kecenderungan sikap penolakan atau penerimaan terhadap aplikasi melalui kegiatan survei dapat memberikan masukan untuk penyempurnaan aplikasi dikemudian hari. Survei dilakukan untuk mendapatkan umpan balik atas persepsi pengguna terhadap Aplikasi. Manfaat Aplikasi diukur berdasarkan parameter-parameter kepuasan pengguna (*User Satisfaction*) terhadap aplikasi.

Manfaat dari melakukan sebuah survey menurut [44] adalah mengukur tingkat kepuasan dari para pengguna agar bisa mendapatkan perhitungan dan kalkulasi yang tepat mengenai fitur-fitur pada aplikasi, dan informasi-informasi yang dibutuhkan adalah tergantung pada pertanyaan dalam survei tersebut. Dari jurnal tersebut [44], dapat ditarik kesimpulan bahwa jenis pertanyaan pada survei yang akan diberikan mempengaruhi hasil perhitungan mengenai fitur-fitur pada aplikasi kami.

Sebagai tanggapan atas pertanyaan-pertanyaan yang telah di berikan, pengguna mungkin akan menemukan pemikiran yang mereka anggap sebagai masalah tambahan yang perlu dipertimbangkan, seperti yang dijelaskan pada [45]. Dengan begitu, hasil survei tidak dimaksudkan untuk memberikan gambaran representatif tentang opini 'kebenaran' terhadap pengalaman pengguna dalam menggunakan aplikasi kami, namun lebih memungkinkan kami membuat sketsa gambaran kasar yang memberikan masukan untuk analisis kualitatif lebih lanjut terhadap penjelasan pengguna untuk mendapatkan jawaban pengguna/responden.

1. **Perhitungan Survei dengan Menggunakan Metode ANOVA**

Analysis of Variance atau ANOVA adalah teknik statistik yang digunakan untuk membandingkan kelompok pada kemungkinan perbedaan rata-rata (mean) dan pengukuran kuantitatif (interval atau rasio, terus menerus). Perhitungan ANOVA sama dengan paired *t-test* untuk membandingakn rata-rata dua sampel yang saling berhubungan. Perbedaan ANOVA dengan *t-test* adalah sampel uji *t-test* adalah sampel pengukuran berulang, sementara ANOVA mensyaratkan sampel bebas [46]. Seperti pada [47], diketahui bahwa variabel yang mengalokasikan responden atau pengguna ke berbagai kelompok disebut faktor; ANOVA dapat melibatkan satu faktor *one-way* (desain satu arah) atau beberapa faktor *two-way* / *multi-way* (rancangan multi arah atau faktorial). Kelompok kami menggunakan design *one-way factor* untuk menghitung rata-rata dari hasil survey. *One-way* ANOVA biasanya digunakan untuk membandingkan nilai disain sebelum dan sesudah dengan partisipan/pengguna yang sama pada satu grup [46].

Melakukan perhitungan survey dengan menggunakan metode ANOVA *one-way* ini sangat membantu karena perhitungan rata-rata survey menjadi lebih terperinci sehingga kami bisa mengetahui kelebihan dan kekurangan dari fitur-fitur pada aplikasi kami.

1. **Survey**

Kami melakukan dua kali survey kepada user yaitu survey sebelum fitur-fitur pada aplikasi *WhatsCamp* kami di aplikasikan dan sesudah aplikasi kami siap digunakan oleh user. Hasil survey sebelum pengaplikasian fitur-fitur pada aplikasi dinamakan survey 1 dan hasil survey sesudah aplikasi siap digunakan dinamakan survey 2.

Survey 1 dilakukan agar kami bisa mengetahui apa saja yang dibutuhkan oleh user pada fitur-fitur kami dengan gambar dari aplikasi lain sebagai contoh dan patokan yang bisa mempermudah penerapan aplikasi *WhatsCamp* kepada *user*.

Survey 2 dilakukan agar kami bisa mengetahui tingkat kenyamanan dan kemudahan *user* dalam menggunakan aplikasi *WhatsCamp* kami.

1. **Hasil Survey 1 (aplikasi sebelum siap digunakan)**

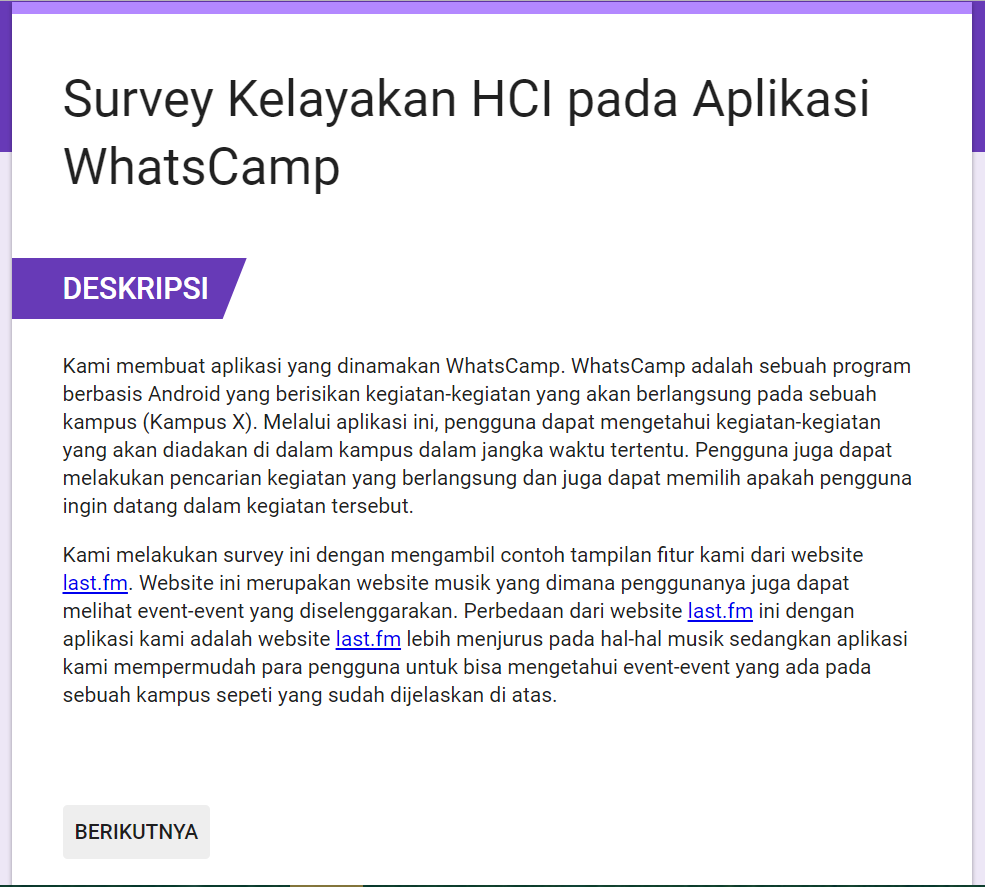
Kelompok kami telah melakukan survei via online dengan mengajukan empat pertanyaan mewakili keempat fitur dari aplikasi kami seperti yang ada pada gambar di bawah ini:

Figure 1. Survey Kelayakan HCI 1.

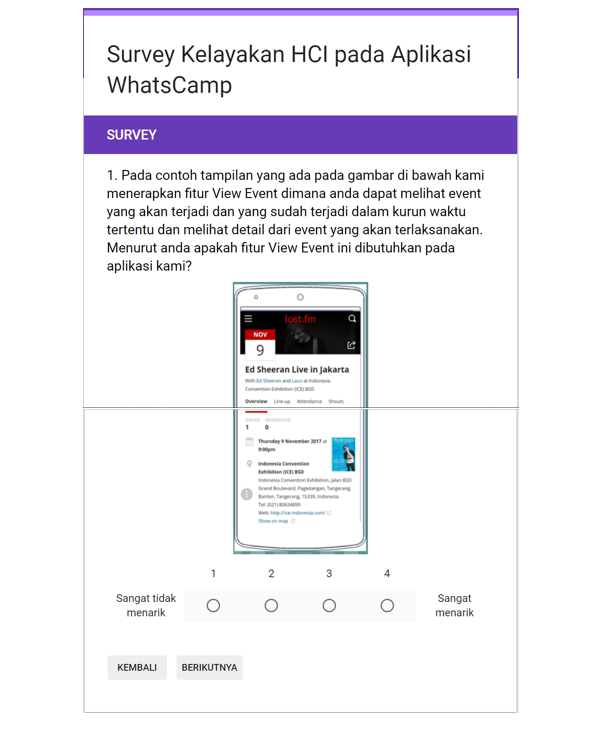


Figure 2. Survey Kelayakan HCI 2.



Figure 3. Survey Kelayakan HCI 3.



Figure 4. Survey Kelayakan HCI 4.

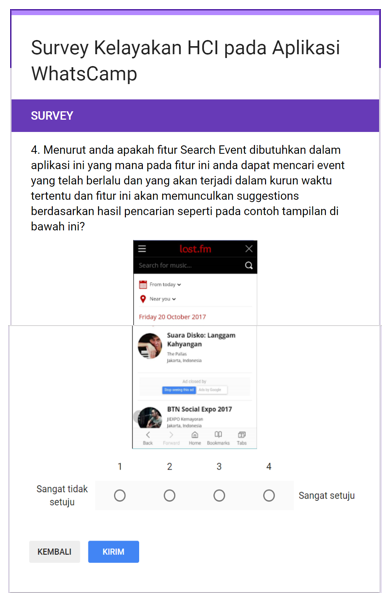


Figure 5. Survey Kelayakan HCI 5.

Setelah kami melakukan survey diatas, kami kemudian melakukan perhitungan pada survey tersebut dengan menggunakan metode ANOVA sehingga mendapatkan hasil dan kesimpulan sepertiyang ada di bawah ini:

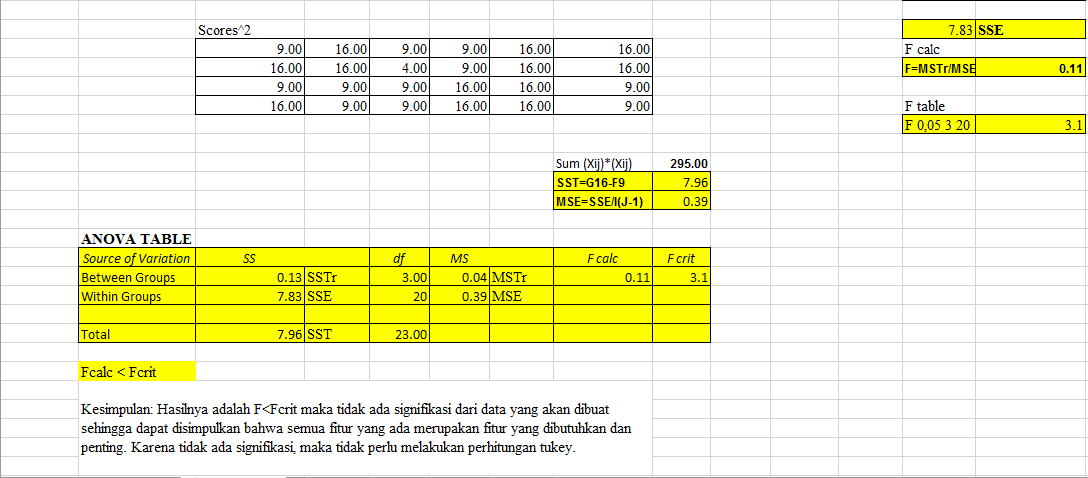
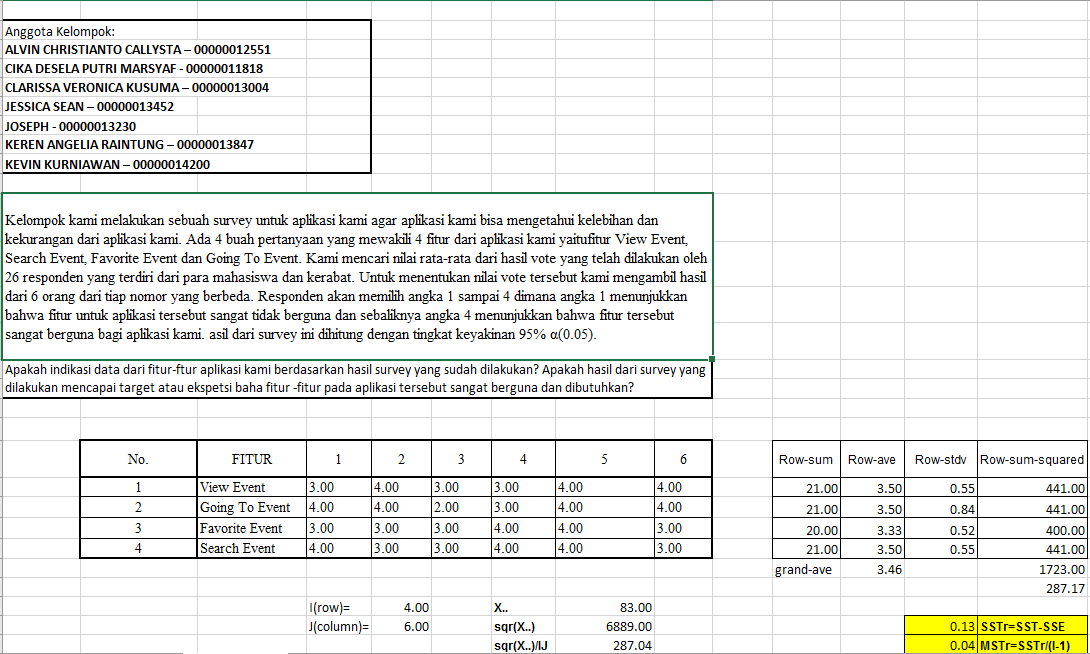


Figure 6. Perhitungan ANOVA SURVEY 1.

1. **Hasil Survey 2 (aplikasi sebelum siap digunakan)**

Sama seperti Survey 1, kelompok kami melakukan survey kembali via online dengan mengajukan empat pertanyaan mewakili keempat fitur dari aplikasi kami seperti yang ada pada gambar di bawah ini:

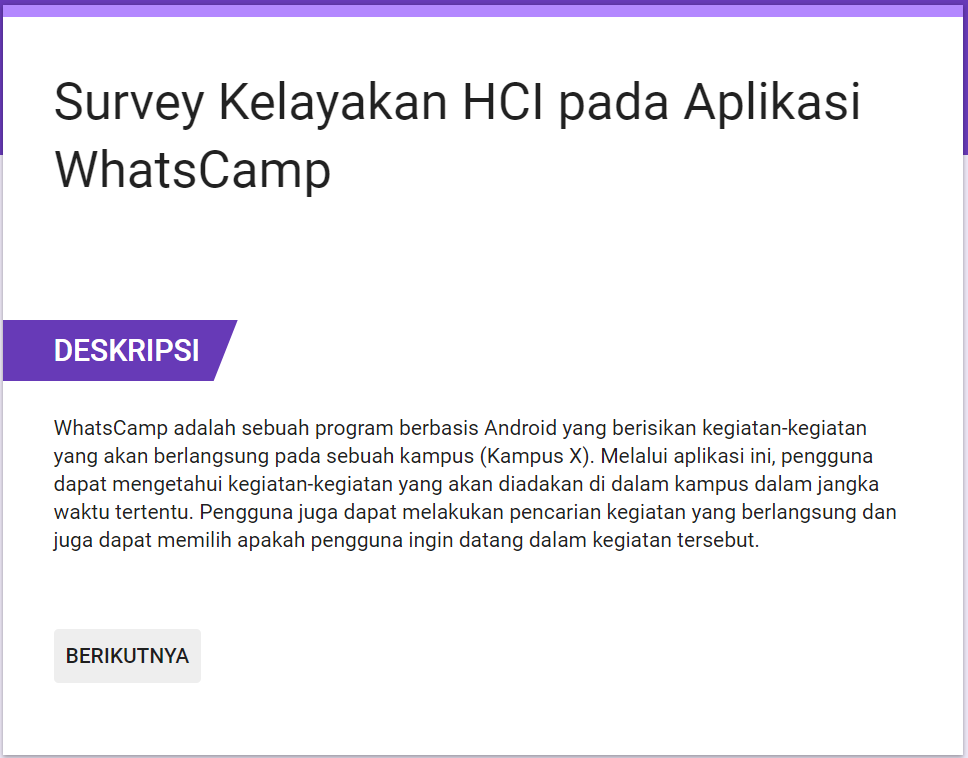
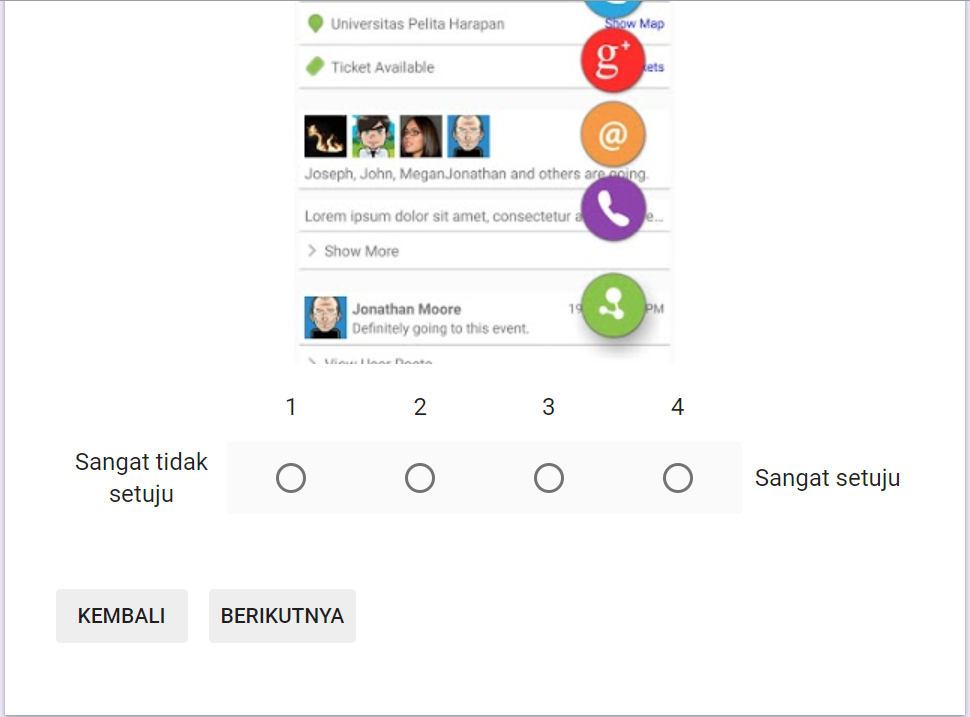


Figure 7. Survey Kelayakan HCI 6.

Figure 8. Survey Kelayakan HCI 7



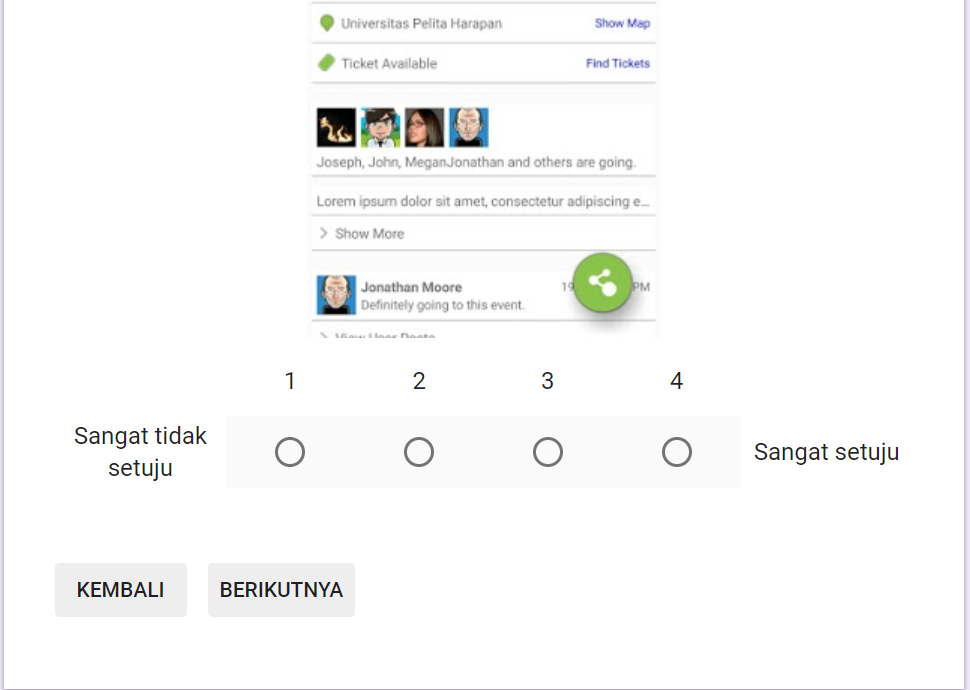


Figure 9. Survey Kelayakan HCI 8.

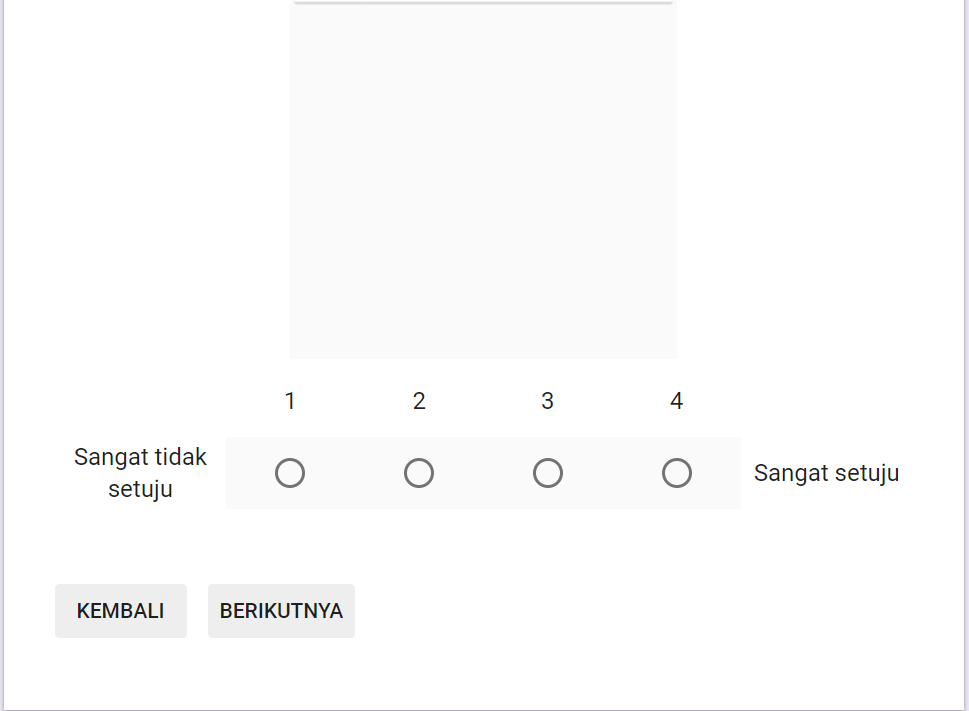


Figure 10. Survey Kelayakan HCI 9

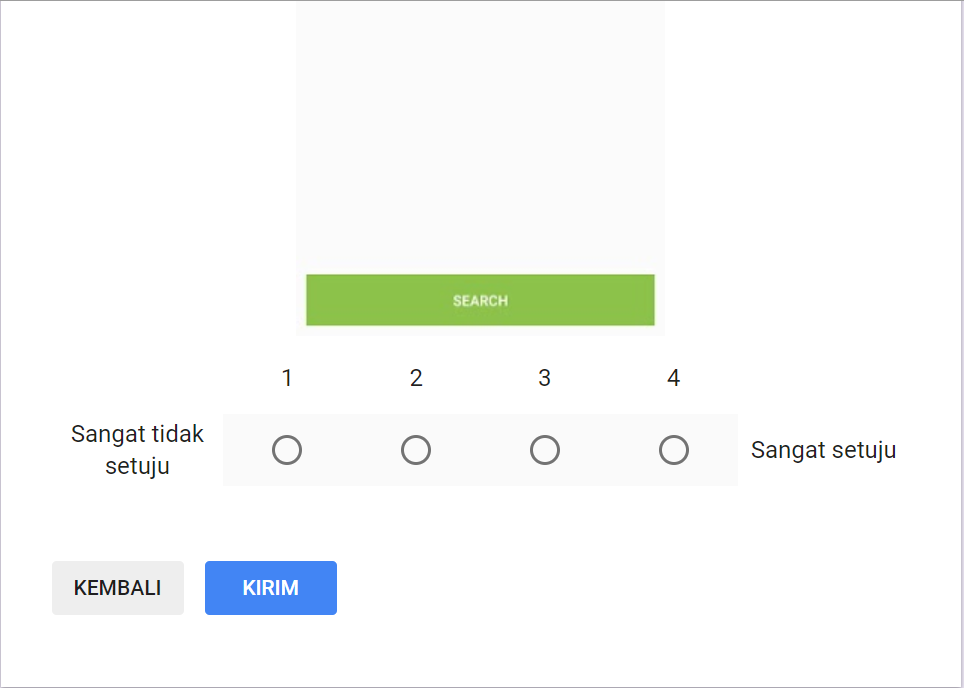


Figure 11. Survey Kelayakan HCI 10.

Setelah kami melakukan Survey 2 kami melakukan perhitungan hasil survey dengan menggunakan metode ANOVA dan TUKEY seperti yang ada pada Gambar 12. Perhitungan ANOVA SURVEY 2 di bawah ini:

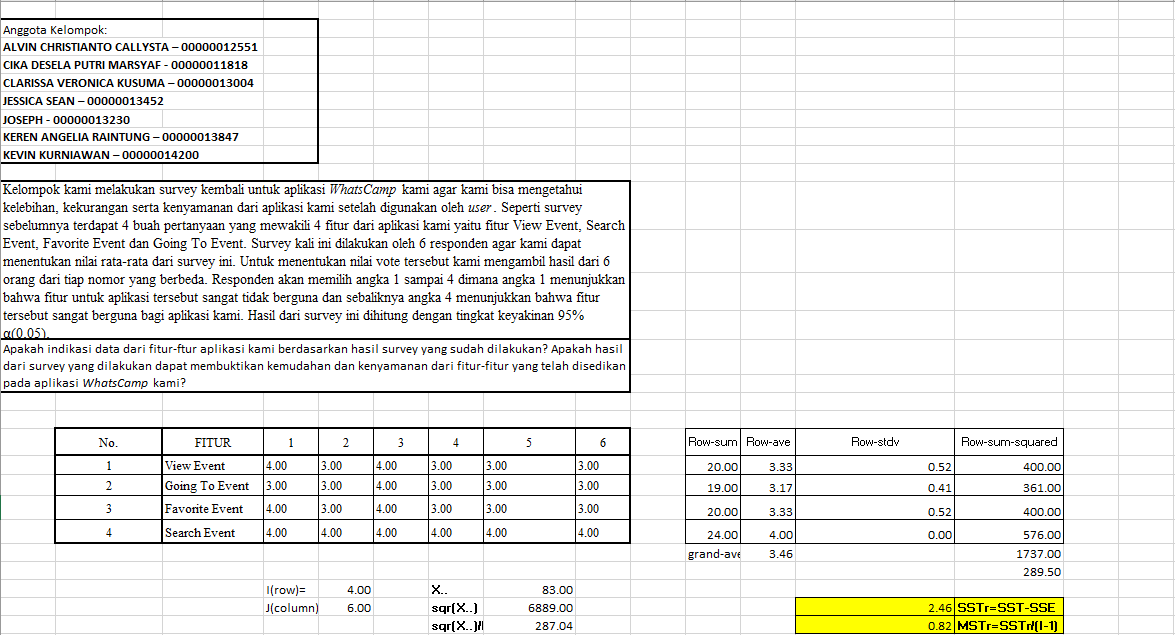
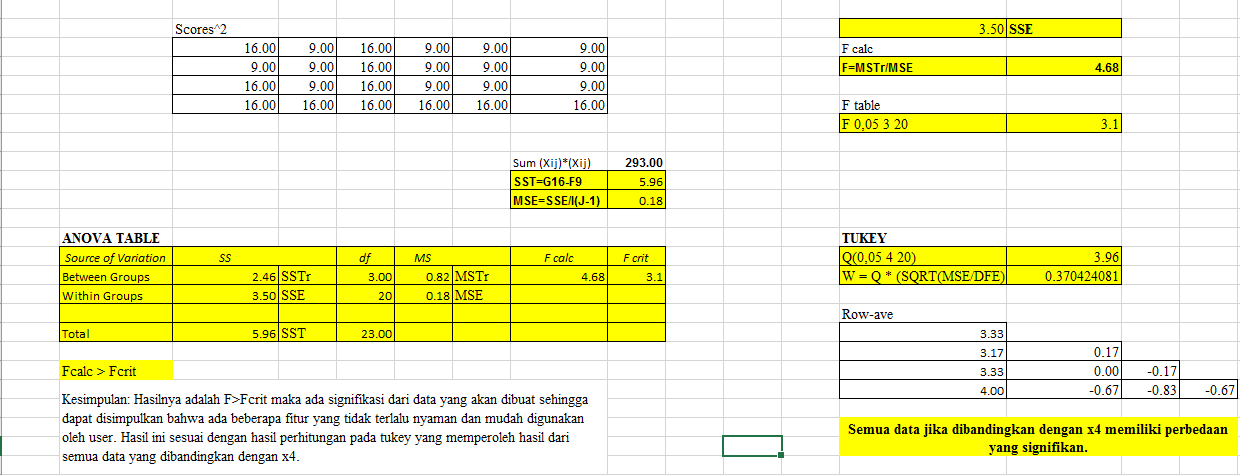


Figure 12. Perhitungan ANOVA SURVEY 2

1. **KESIMPULAN**

Jadi, kesimpulan dari survey yang telah dilakukan adalah pada Survey 1, hasil perhitungan ANOVA dengan tingkat keyakinan 95% α (0.05) mendapatkan hasil yang tidak signifikan yaitu F<Fcrit sehingga kami tidak perlu melakukan perhitungan dengan menggunakan TUKEY dan hasil tersebut membuktikan bahwa fitur-fitur yang akan ada pada aplikasi kami semuanya dibutuhkan dan penting, sehingga fitur View Event, Search Event, Going To Event dan Favorite Event akan diterapkan dalam aplikasi kami yaitu WhatsCamp.

Pada Survey 2, hasil perhitungan pada ANOVA dengan tingkat keyakinan 95% α (0.05) mendapatkan hasil yang signifikan yaitu F>Fcrit sehingga perlu dilakukan perhitungan pada TUKEY yang membuktikan bahwa semua data jika dibandingkan dengan x4 memiliki perbedaan yang signifikan dalam artian ada beberapa fitur pada aplikasi kami yang kurang nyaman dan masih perlu dikembangkan lagi sehinga lebih mudah untuk digunakan oleh *user*.

# BAB II

# FEASIBILITY STUDY

## DATABASE – Jessica Sean

* + - 1. Deskripsi Fitur

Aplikasi yang kita rancang merupakan aplikasi yang memberikan pengguna informasi mengenai kegiatan dan *event* yang sedang berlangsung di dalam kampus. Dengan demikian, aplikasi ini perlu didukung untuk fitur database. Database dalam aplikasi ini digunakan untuk menyimpan data mengenai informasi-informasi kegiatan dan *event* yang termasuk di dalamnya informasi kegiatan, tanggal dan waktu pelaksanaan, dan lain sebagainya. Database juga berguna untuk menunjukkan hasil *query* *filtering* event berdasarkan kategori dan jarak dan *query* lainnya, serta kemampuan untuk melakukan *update* terhadap *event-event* tanpa perlu melakukan update pada aplikasi secara keseluruhan. Dengan adanya *update* yang berkala, diharapkan aplikasi ini terus akan menampilkan *event* kampus yang terbaru.

* + - 1. Resiko

Untuk mengimplementasi sebuah database yang dapat memenuhi fitur yang telah dideskripsikan diatas, perlu dilakukan beberapa pertimbangan terhadap cara untuk menyimpan data di dalam *mobile* phone. Menurut journal yang berjudulkan “*Centralized Database for Android and Web Application*” ada beberapa jenis cara untuk menyimpan data relasi dalam mobile phone, yaitu:

* XML/JSON *Local Storage*
* Database *Local* (*Database management systems*)
* *Remote* Database (Web) [24]

Ketiga cara diatas memiliki kelebihan dan kekurangan masing-masing yang akan dirincikan di bawah ini:

* + - 1. XML/JSON *Local Storage*

Cara ini merupakan cara yang paling mudah, namun hasil implementasi XML/JSON *Local Storage* ini tidak diharapkan. Hal ini dikarenakan aplikasi kita memerlukan *update* database yang berkala terutama ketika ada *event* diadakan di kampus. Sedangkan solusi menyimpan data menggunakan XML/JSON Local Storage tidak mengijinkan adanya update pada database saja. Sehingga, ketika data di dalam XML/JSON Local Storage ingin diubah dan diupdate, siperlukan update secara keseluruhan aplikasi. Selain itu, XML/JSON Local Storage juga tidak menyediakan unsur *security* dan integritas data.

Berikut merupakan contoh penyimpanan data menggunakan JSON File [13]:

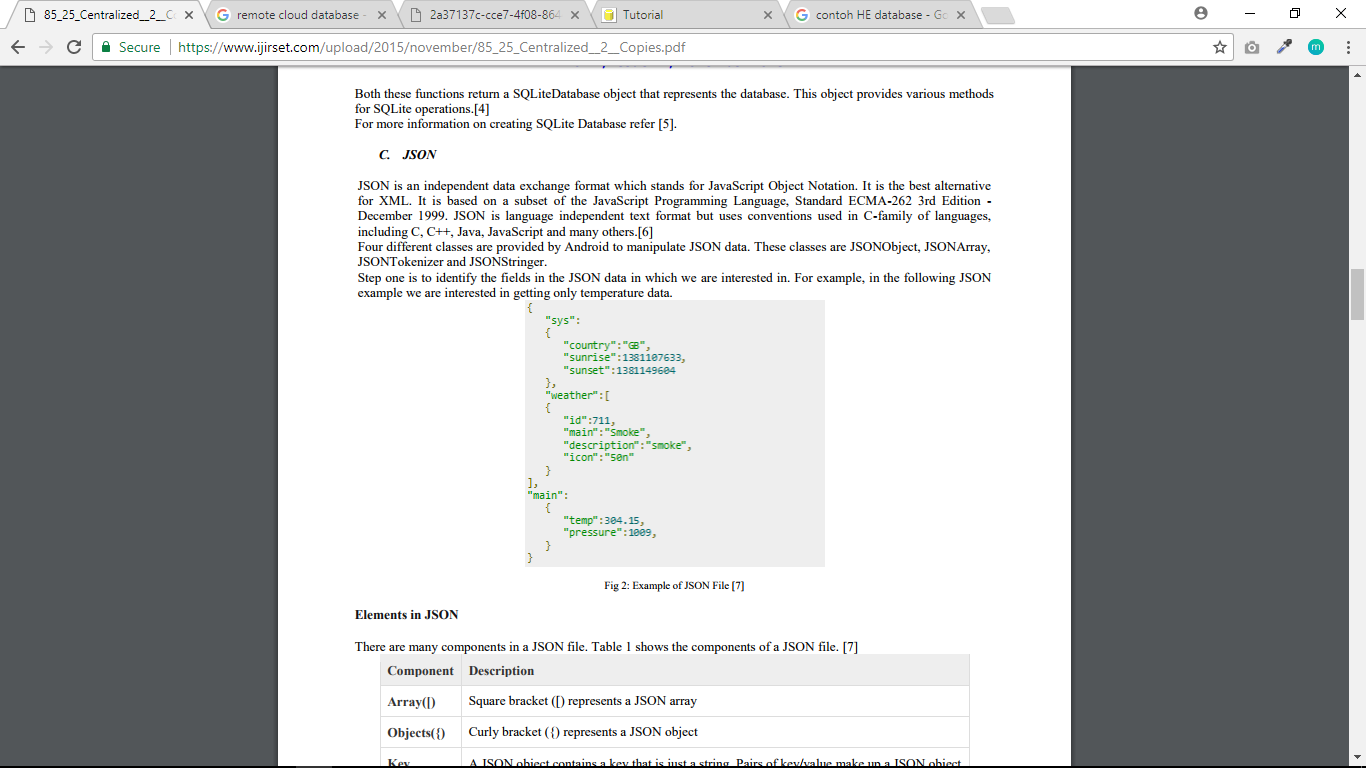


Figure 13. Penyimpanan Data Menggunakan JSON File. [13]

* + - 1. Database *Local* (*Database management systems – H2/SQLite*)

Untuk membangun sebuah aplikasi mobile menggunakan *Database management systems*, perlu untuk memperhatikan kedua faktor, yaitu optimasi terhadap *power* dan manajemen memori [25]. Pertama, *power optimation* merupakan faktor yang penting karena perangkat seperti *smartphone* memiliki *power supply* yang terbatas. Kedua, hal yang harus dipertimbangkan merupakan besarnya memori yang dapat digunakan dalam satu waktu. Oleh karena itu, optimasi terhadap memori (*memory management*) dibutuhkan di dalam sebuah aplikasi mobile. Hal-hal lain yang perlu dipertimbangkan adalah OS, *acidity,* konten dan ukuran maksimal database, tabel, dan ukuran properti lainnya [25]. Karena aplikasi ini dijalankan pada OS Android dan data yang digunakan dalam aplikasi ini berupa *relational* data, maka ada 2 pilihan *database management system* yang dapat digunakan, yaitu SQLite (database yang ringan) dan H2 (database yang kompleks).

Kami mempertimbangkan beberapa hal untuk memilih DBMS yang akan kami gunakan. Jika kita memilih menggunakan H2, pertimbangannya adalah kemungkinan waktu untuk mengakses *query* akan lebih lama. Tetapi, kelebihan dari H2 adalah ketersediaanya terhadap hampir seluruh kebutuhan untuk mengakses data. Sedangkan kelebihan dari SQLite merupakan waktu mengakses *query* yang lebih cepat dan kekurangannya adalah kebutuhan untuk mengakses data yang cukup terbatas [26].

Selain itu, berhubung dengan aplikasi ini dijalankan di mobile dan terdapat beberapa fitur yang dapat digunakan di dalam aplikasi kita, maka project ini tidak terbatas hanya pada satu atau dua *query* saja. Sehingga, untuk mengimplementasi database yang kompleks seperti H2 akan membutuhkan waktu yang cukup banyak.

1. *Remote* Database (*Web*)

Cara yang paling sulit, namun yang paling baik untuk menyimpan data yang digunakan untuk aplikasi ini adalah *Remote Database*. Dengan menggunakan *remote database*, perlu dibuat *hosting/webserver* yang meng-*host* database, sehingga *update* hanya dapat dilakukan ketika *device* terkoneksi dengan internet. Selain itu, untuk menggunakan *remote control*, diperlukan pengetahuan yang baik terhadap *computer security* untuk membuat sistem sekuriti dari database sehingga data tidak dapat dimanipulasi oleh *third-party*.

* + - 1. Penyelesaian Resiko

Dalam memilih cara menyimpan data yang ada, kami memilih cara yang sesuai dengan syarat-syarat yang telah disebutkan di bagian deskripsi fitur database. Selanjutnya, di dalam mengimplementasi database yang telah kami pilih, kami juga akan melakukan beberapa modifikasi terhadap bentuk database yang kita pilih sehingga hasil yang optimal dari mengimplementasi bentuk database tersebut dapat diperoleh. Sebagai contoh, kami dapat menggunakan berbagai *plugin* yang dapat menambah fungsi-fungsi yang diperlukan.

* + - 1. Kesimpulan

Berdasarkan pertimbangan yang dilakukan oleh kelompok kami, maka kami memutuskan untuk menggunakan SQLite. Hal ini dikarenakan SQLite cukup baik untuk memenuhi syarat-syarat yang telah disebutkan pada deskripsi fitur database, terutama syarat untuk melakukan *update* database tanpa melakukan *update* terhadap keseluruhan aplikasi. Selain itu, karena keterbatasan waktu dalam mengerjakan *project*, maka menurut kami solusi yang paling memungkinkan dan baik adalah menggunakan SQLite. Jika waktu memungkinkan, maka kelompok kami akan berusaha untuk mencoba mengimplementasikan *remote* database, yang merupakan solusi paling elegan.

# 

## DESIGN – Clarissa, Joseph, Kevin Kurniawan

* + - 1. Deskripsi Fitur

Aplikasi WhatsCamp merupakan aplikasi *mobile* yang memiliki interaksi langsung dengan *end-user*. Maka dari itu, diperlukan penerapan *design* yang sederhana dan mudah untuk digunakan sehingga dapat memfasilitasi keinteraktifan antara pengguna dan aplikasi, serta dapat meningkatkan keefektifan dari apa yang dilakukan pengguna [27]. Pada bagian ini, akan dibahas mengenai *design*, mungkinkah untuk diimplementasi, resiko yang mungkin ada, dan bagaimana cara menangani resiko tersebut. Pembahasan *design* ini juga berdasarkan pada pembahasan *User Interface*, Unsur Estetika dalam *User Interface*, Hubungan HCI dengan *User Interface* pada bagian *Literature Survey* sebelumnya. Pembahasan bagian ini juga merujuk pada perancangan *user interface* dari aplikasi yang akan kami buat.

Design *user interface* yang kami maksudkan pada bagian ini meliputi komponen-komponen sebagai berikut.

1. Perancangan tampilan tata letak (*layout*) sebagai tempat penempatan komponen-komponen lainnya.
2. Nama aplikasi.
3. Menu, untuk mengakses *view* lainnya.
4. Tombol, untuk menjalankan aktivitas tertentu.
5. Termasuk pemilihan *properties* dari komponen-komponen yang ada, seperti grafis, warna, bentuk, ukuran, dan aspek *properties* lainnya.

Di samping itu, untuk dapat membuat *design* yang mudah dimengerti, lebih menarik, dan efisien, maka prinsip *Human Computer Interaction* (HCI) [5], sebagaimana yang telah dibahas pada bagian Hubungan HCI, prinsip *User Interface* [1], dan metode-metode pada bagian Unsur Estetika dalam *User Interface* [8] pada bagian *Literature Survey* sebelumnya, juga perlu untuk diterapkan dalam perancangan ini. Hal tersebut bertujuan agar perancangan tidak hanya fokus dalam membuat tampilan aplikasi menjadi bagus saja, melainkan juga fokus dalam membuat aplikasi yang lebih interaktif dengan pengguna sehingga mempermudah penggunaannya.

Berikut merupakan contoh aplikasi *mobile* lain sebagai gambaran dari perancangan *user interface* dari aplikasi yang akan kami buat.

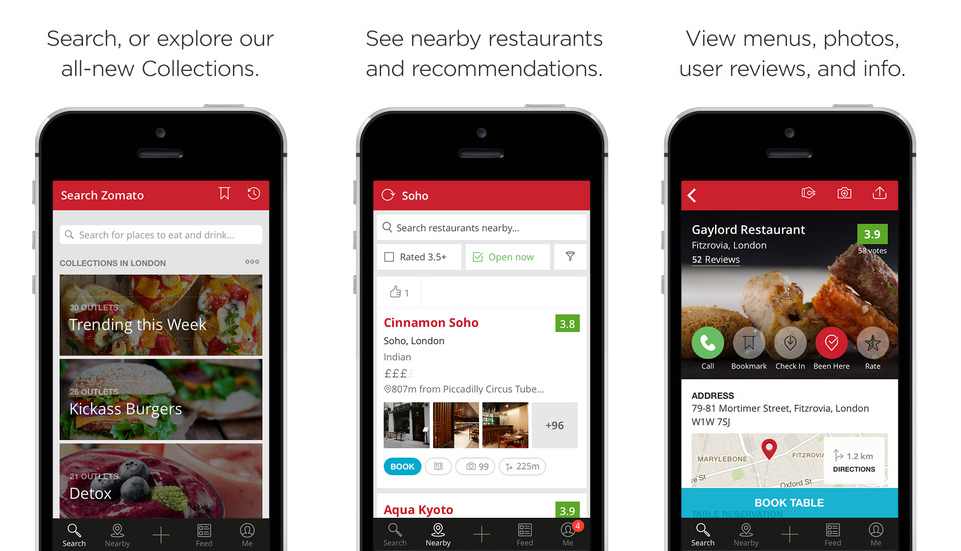


Figure 14. Contoh User Interface. [28]

Aplikasi di atas merupakan salah satu aplikasi pencarian restoran yang menyediakan informasi mengenai berbagai restoran di seluruh dunia. Aplikasi tersebut telah diunduh lebih dari 274.000 kali dan memiliki ulasan 4.3/5 bintang. Aplikasi tersebut memberikan informasi mengenai tempat, menu, harga, dan foto-foto yang dapat diunggah oleh pengguna yang pernah berkunjung ke restoran tertentu. Selain itu, pengguna juga dapat memiliki interaksi langsung terhadap konten di dalamnya dengan memberikan ulasan mengenai pengalamannya berkunjung ke restoran tersebut. [29]

Menurut kami, aplikasi ini cukup interaktif, mudah dimengerti, dan menyediakan apa yang menjadi kebutuhan pengguna secara jelas. Aplikasi ini memiliki perancangan tata letak yang rapi, pilihan menu yang jelas, *icon-icon* yang mudah dikenali, serta *properties* grafis dari setiap bagian yang senada dengan keseluruhan aplikasi. Secara umum, aplikasi ini cukup baik dalam menerapkan prinsip-prinsip dari *User Interface* dan prinsip-prinsip dari HCI sehingga aplikasi ini pun cukup diminati oleh banyak pengguna.

1. Resiko

Resiko yang mungkin ada dalam perancangan *user interface* dari aplikasi kami adalah sebagai berikut.

1. Kesulitan untuk mengetahui apakah perancangan dari aplikasi kami sudah memenuhi kriteria yang diharapkan oleh sebagian besar pengguna.
2. Kesulitan untuk mengetahui apakah perancangan yang kami terapkan sudah sesuai dengan standar yang umumnya digunakan sebagaimana dibahas pada bagian sebelumnya, dan dengan dasar-dasar atau prinsip yang digunakan dalam pembuatan *user interface* [1] sebagaimana telah dibahas pada bagian *User Interface* dalam *Literature Survey*, metode-metode sebagaimana telah dibahas pada bagian Unsur Estetika dalam *User Interface* [8], dan prinsip-prinsip HCI [5] sebagaimana telah dibahas pada bagian Hubungan HCI dengan *User Interface* dalam *Literature Survey*.
3. *Design* dari aplikasi yang kami buat tidak dimengerti oleh pengguna.
4. Kesulitan untuk menyelesaikan *design* aplikasi sesuai dengan waktu yang telah ditetapkan sebelumnya sehingga dapat mengganggu proses lainnya.
5. Penyelesaian Resiko

Berikut merupakan hasil diskusi kami untuk menangani beberapa resiko yang telah disampaikan pada bagian sebelumnya.

1. Untuk mengetahui kriteria yang diharapkan oleh sebagian besar pengguna, maka kami akan melakukan survei yang akan diisi oleh pengguna dan membuka kemungkinan adanya perubahan dari *design* kami yang sebelumnya dengan tetap mempertimbangkan batasan waktu yang ada.
2. Kurangnya pengetahuan kami akan prinsip-prinsip atau standar yang umumnya digunakan, memungkinkan kami untuk kembali ke tahap *design* meskipun kami telah berada di tahap selanjutnya. Jika kami menemukan kekurangan/kesalahan saat melakukan pengimplemetasian *design*, maka kami akan kembali ke tahap *design* tersebut dengan tetap menyesuaikan batasan waktu yang ada.
3. Walaupun *design* yang dibuat telah dirancang sedemikian rupa, kami sadari bahwa tidak menutup kemungkinan adanya pengguna yang tidak mengerti dalam menggunakan aplikasi kami. Maka dari itu, kami akan membuat panduan penggunaan aplikasi (*user manual*) dengan tujuan untuk membantu pengguna yang mengalami kesulitan tersebut.
4. Untuk permasalahan penyelesaian design aplikasi lebih dari waktu yang telah ditetapkan, maka kami memutuskan agar dalam proses *design*, kami fokus hal-hal esensial yang harus dipastikan terlebih dahulu (seperti penyusunan tata letak). Setelah itu, kami baru masuk ke hal yang lebih detail (seperti *background*, warna tombol, ukuran *font*), sehingga jika batas waktu *design* telah sampai, maka kami dapat langsung pindah ke tahapan selanjutnya.
5. **Kesimpulan**

Melihat beberapa resiko yang telah kami paparkan pada bagian sebelumnya beserta hasil diskusi kami mengenai cara penanganannya, maka kami memutuskan untuk tetap menerapkan *design* dalam aplikasi yang kami buat. Penerapan *design* ini tentunya mengutamakan dasar-dasar atau prinsip [27] sebagaimana yang telah dipaparkan dalam pembahasan *Literature Survey* bagian *design* dan tetap mempertimbangkan batasan waktu yang telah ditentukan sebelumnya. Jika hasil penerapan *design* belum sempurna pada waktu yang ditentukan, maka kami tetap akan melanjutkan ke tahapan selanjutnya.

## MAIN FEATURE - Cika Desela, Alvin Callysta, Keren Angelia

1. **Deskripsi Fitur**

Pada bagian ini akan dibahas mengenai *main feature* dari aplikasi *WhatsCamp* yang kami rancang, apa saja resiko yang mungkin akan dihadapi dalam pengimplementasian dan bagaimana cara untuk mengatasi masalah tersebut. Kami memiliki 4 buah fitur utama yaitu; *View Event, Favorite Event, Search Event* dan *Going to Event*.

* 1. **Fitur Utama: *View Event***

Fitur ini memiliki fungsi bagi pengguna untuk melihat berbagai eventyang ada di dalam aplikasi rancangan kami, *View Event* memungkinkan pengguna untuk:

* Melihat event yang akan terjadi dan yang sudah terjadi dalam kurun waktu tertentu
* Melihat detail dari event yang akan terlaksanakan.
  1. **Fitur Utama: *Favorite Event***

Fitur ini memiliki fungsi bagi pengguna untuk memfavoritkan event yang sudah terjadi dan yang akan terjadi sehingga pengguna bisa melihat kembali event yang difavoritkan yang kemudian akan ditampilkan di sebuah halaman khusus *Favorite Event*.

* 1. **Fitur Utama: *Search Event***

Fitur ini memiliki fungsi bagi pengguna untuk mencari event yang diinginkan. Fitur ini memungkinkan pengguna untuk mencari event yang telah berlalu dan yang akan terjadi dalam kurun waktu tertentu. Selain itu, fitur *Search Event* akan memunculkan *suggestions* berdasarkan hasil pencarian.

* 1. **Fitur Utama : *Going to Event***

Fitur ini memiliki fungsi bagi pengguna untuk melakukan “informasi kedatangan” untuk menghadiri sebuah event. Jika pengguna melakukan informasi kedatangan, maka akan muncul halaman yang menampilkan berapa banyak orang (pengguna lainnya) yang akan hadir dalam suatu event yang sama. Ketika tanggal acara dari sebuah event semakin dekat, maka akan muncul *reminder* atau notifikasi tentang acara tersebut.



Figure 15. Ilustrasi Fitur yang Dirancang 1. [30]

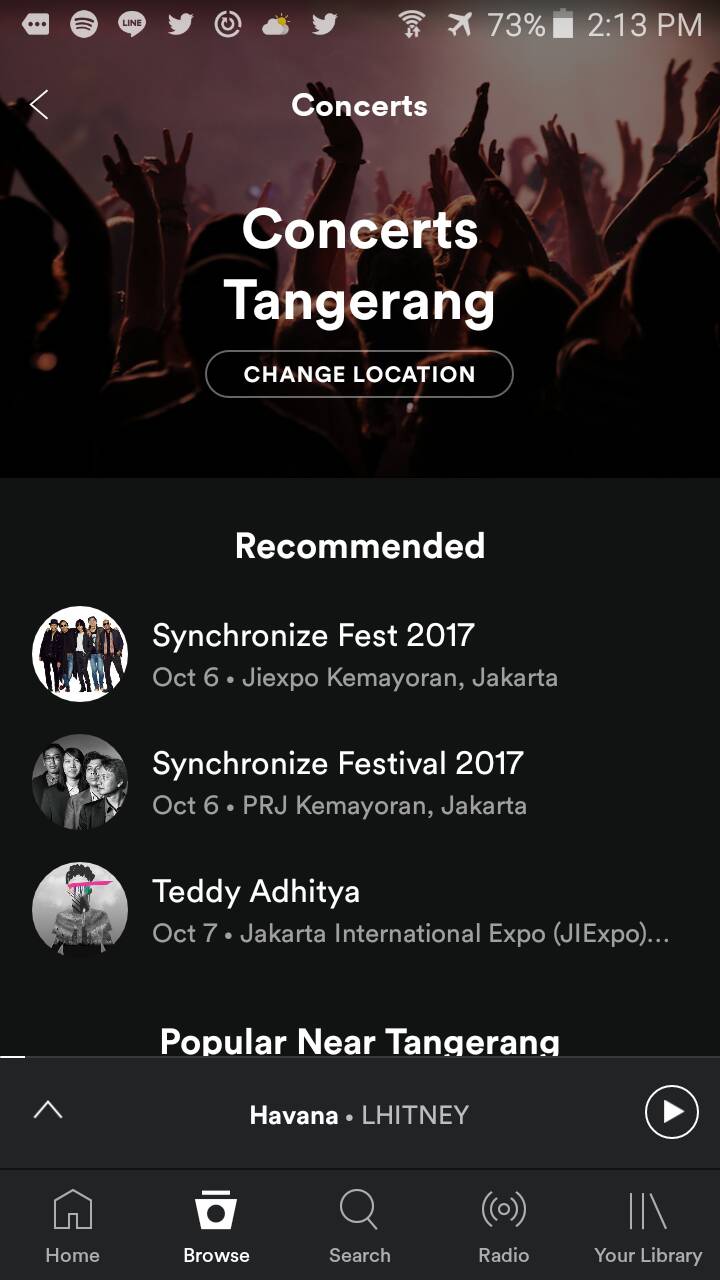


Figure 16. Ilustrasi Fitur yang Dirancang 2. [31]

1. **Resiko**

Resiko yang mungkin akan dihadapi dalam beberapa fitur utama secara umum adalah:

* Dalam *View Event* adanya kemungkinan halaman tidak menampilkan event dengan cepat (proses *retrieving data* yang lama) sehingga membuat pengguna menunggu lama untuk melihat event.
* Dalam *Favorite Event* adanya kemungkinan event yang telah dipilih tidak muncul dalam halaman khusus *Favorite Event.*
* Dalam *Favorite Event* adanya kemungkinan event yang terhapus pada halaman khusus *Favorite Event* apabila event yang telah berlalu sudah melebih batas waktu tertentu.
* Dalam *Search Event* adanya kemungkinan dimana event yang dicari tidak muncul pada tampilan, baik berupa event yang tidak bisa diakses maupun *suggestion* yang tidak relevan sesuai dengan keinginan pengguna.
* Dalam *Search Event* adanya kemungkinan proses pencarian event yang lama yang dapat mengurangi kenyamanan bagi pengguna dalam menggunakan aplikasi.
* Dalam *Going to Event,* memerlukan koneksi internet untuk dapat menampilkan berapa banyak pengguna yang kemungkinan akan hadir dalam event yang sama secara *real-time.* (karena banyaknya pengguna yang akan menghadiri sebuah event dapat berubah sewaktu-waktu).
* Dalam *Going to Event,* adanya kemungkinan pengguna melakukan konfirmasi kedatangan ke sebuah event tanpa disengaja (*human error)*.

1. **Penyelesaian Resiko**

Melalui hasil diskusi dengan kelompok, beberapa solusi yang dapat kami tawarkan untuk mengatasi permasalahan dalam *Main Feature* adalah sebagai berikut:

* Untuk mengatasi permasalahan dalam fitur yang berhubungan dengan database (dari segi pencarian dan informasi data), maka kami akan memastikan bahwa database yang kami buat telah sesuai dengan *Literature Study* dan *Feasibility Study* mengenai database, supaya saat pengimplementasiannya nanti aplikasi dapat berjalan dengan baik dan dapat menampilkan informasi yang sesuai dengan keinginan pengguna dengan tepat dan cepat (efisien).
* Untuk permasalahan *suggestion,* supaya menghasilkan *suggestion* yang relevan dan sesuai dengan keinginan pengguna, kami akan menerapkan algoritma pencarian yang sesuai dengan database yang kami rancang.
* Untuk mengetahui apakah fitur yang kami rancang telah sesuai dengan kebutuhan pengguna maka kami akan melakukan survei kepada pengguna sesuai dengan [32] yang nantinya akan menjadi target *user* dari aplikasi yang kami rancang.
* Untuk mengetahui apakah fitur yang dirancang dapat berjalan dengan baik saat digunakan pengguna, maka kami akan lakukan *testing* pada aplikasi terlebih dahulu dimana *testing* tersebut berdasarkan konsep dasar usabilitas pada *Human-Computer Interaction* [33] dengan metode yang sekiranya sesuai dengan fitur kami [34].
* Untuk permasalahan koneksi internet, kami akan melakukan pengujian dengan cara menyimpan *cache* data aplikasi yang dibutuhkan secara lokal di dalam aplikasi sehingga aplikasi tetap bisa berjalan tanpa adanya koneksi internet [35].
* Untuk permasalahan dalam fitur *Going to Event*, akan dibuat *alert* dialog atau *button* untuk menghindari terjadinya *human error* [21] dan memastikan bahwa pengguna benar-benar melakukan konfirmasi kedatangan ke sebuah event berdasarkan kesadarannya (tanpa disengaja).

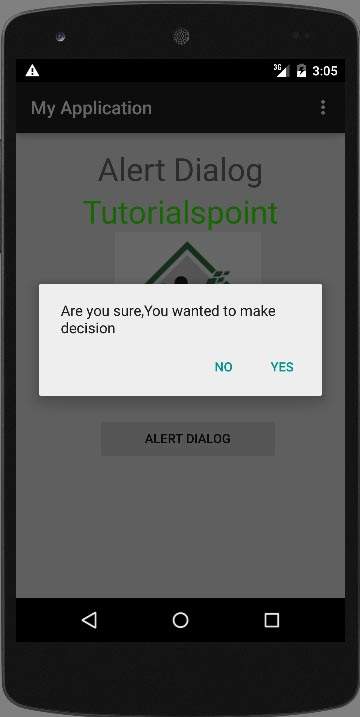


Figure 17. Ilustrasi Alert Dialog. [36]

1. **Kesimpulan**

Dari hasil diskusi, kami telah mempertimbangkan resiko yang mungkin muncul beserta solusi yang dapat kami tawarkan, kami memutuskan untuk menerapkan *Main Feature: View Event, Search Event, Favorite Event,* dan *Going to Event* dalam aplikasi *WhatsCamp* yang kami buat. Penerapan *Main Feature* ini tidak terlepas dari hasil *Literature Study* untuk mengimplementasikan fitur yang kami rancang. Namun, tidak menutup kemungkinan akan terjadi adanya perubahan dalam *Main Feature* ini dikarenakan keterbatasan waktu pengerjaan aplikasi maupun kendala lainnya.

# BAB III

# DESAIN APLIKASI

## DESAIN AWAL – Kevin Kurniawan

Pada pembuatan aplikasi WhatsCamp, kami menggunakan *template* yang bernama EventFinder [37] yang kami dapatkan dari codelist.cc. Kami memilih template EventFinder [37] ini sebagai *template* dasar aplikasi kami karena menu dan fungsi dari template ini sendiri sesuai dengan aplikasi yang kami rencanakan, baik *layout* maupun menu dari fitur-fitur yang akan kami implementasikan. Berikut merupakan *screenshot* dari aplikasi EventFinder [37] sebelum kami melakukan modifikasi.



*Figure 18. Splash Screen. [37]*

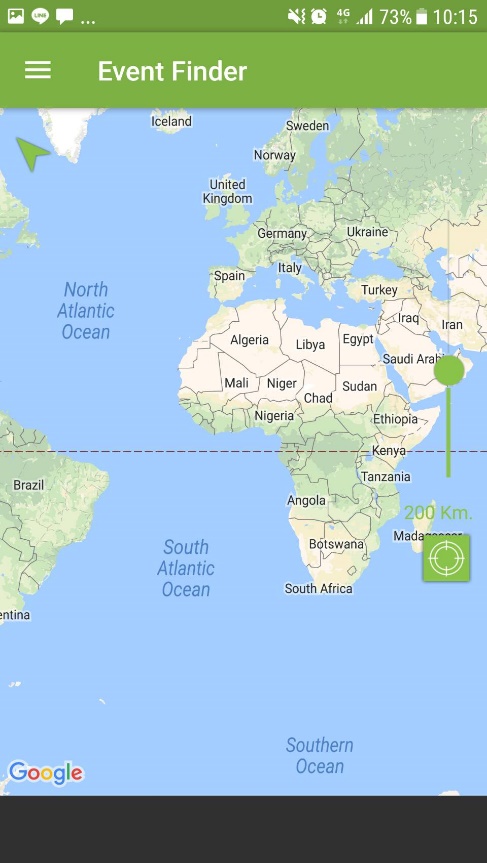


Figure 19. Menu Home. [37]



Figure 20. Menu Aplikasi (Atas). [37]

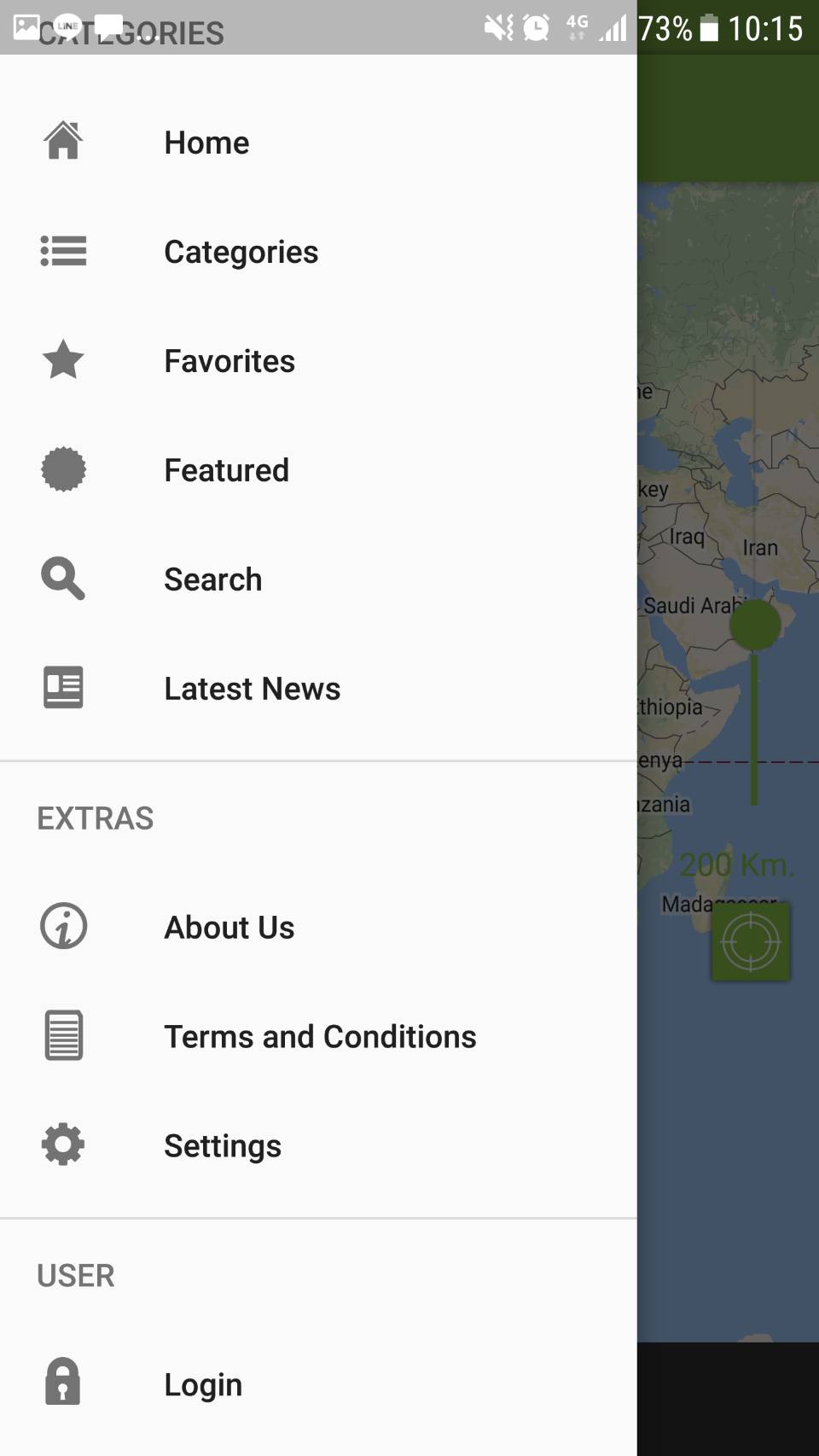


Figure 21. Menu Aplikasi (Bawah). [37]

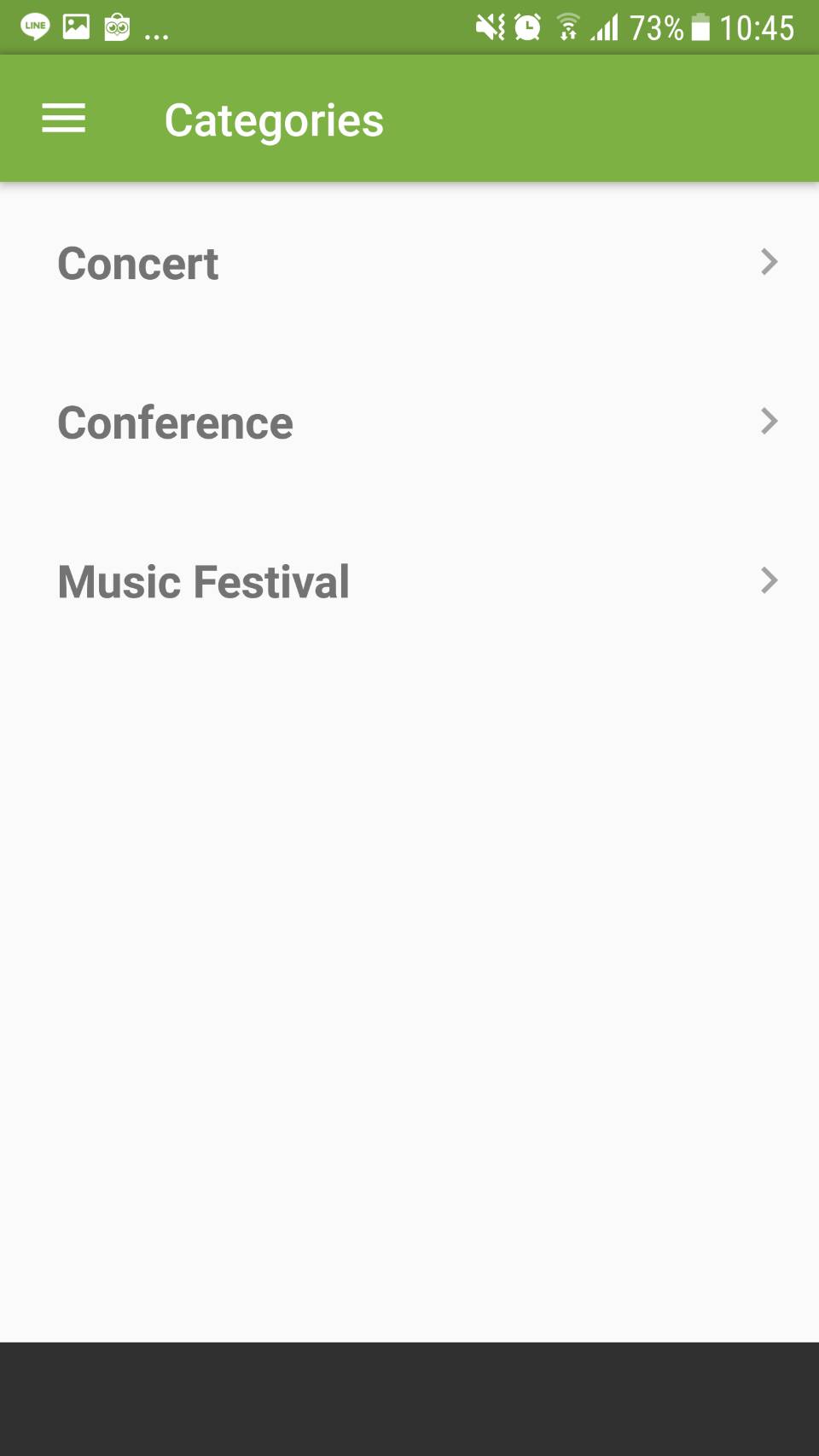


Figure 22. Menu Categories. [37]

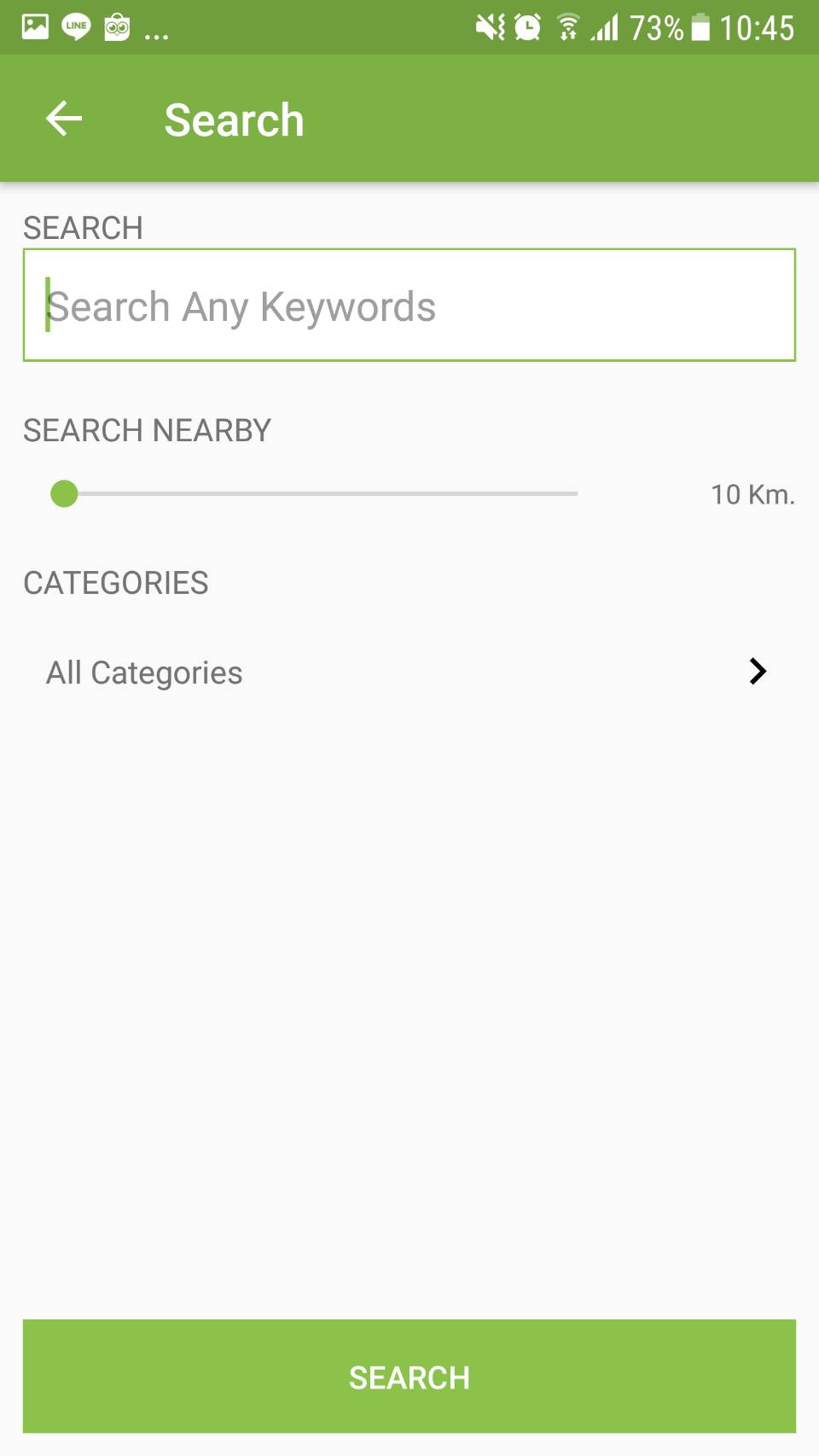


Figure 23. Menu Search. [37]

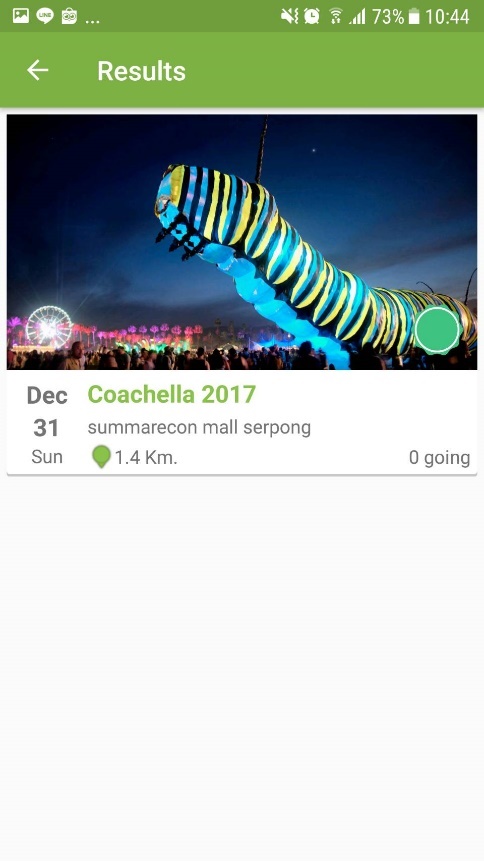


Figure 24. Halaman Hasil Pencarian. [37]

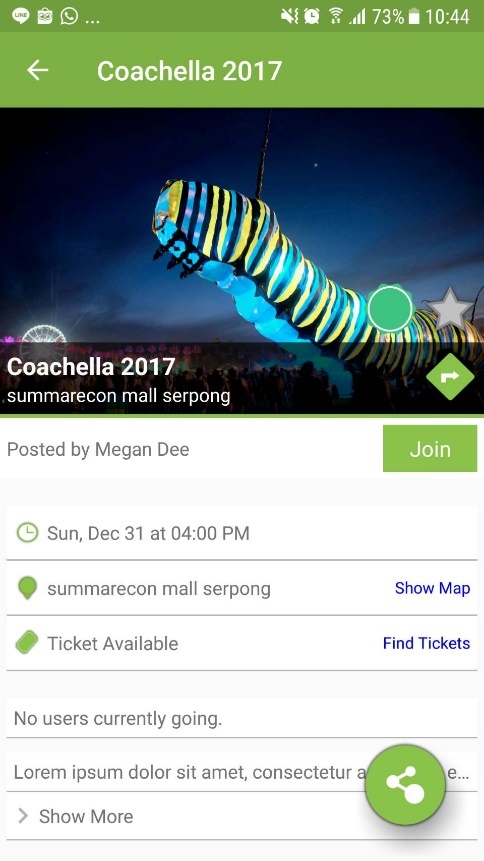


Figure 25. Halaman Event. [37]

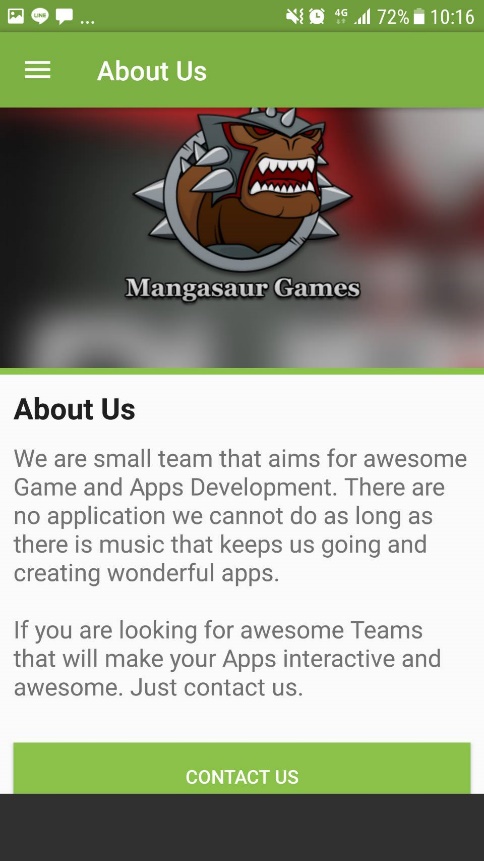


Figure 26. Menu About Us. [37]

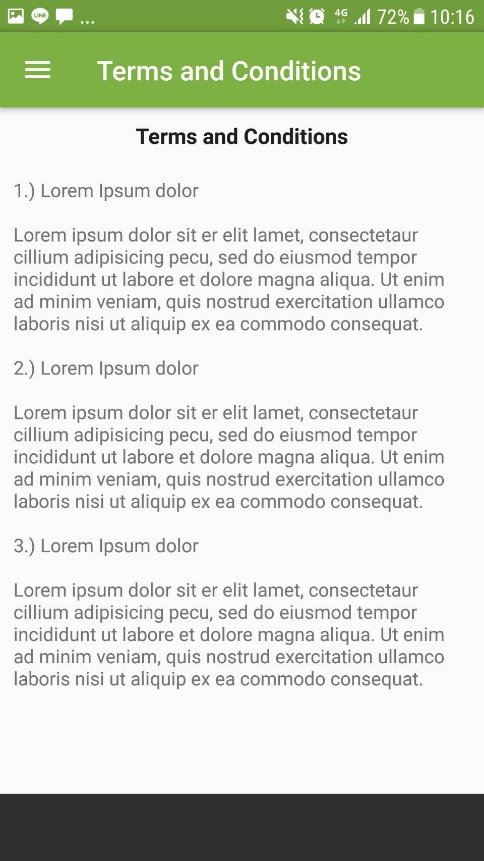


Figure 27. Menu Terms and Conditions. [37]

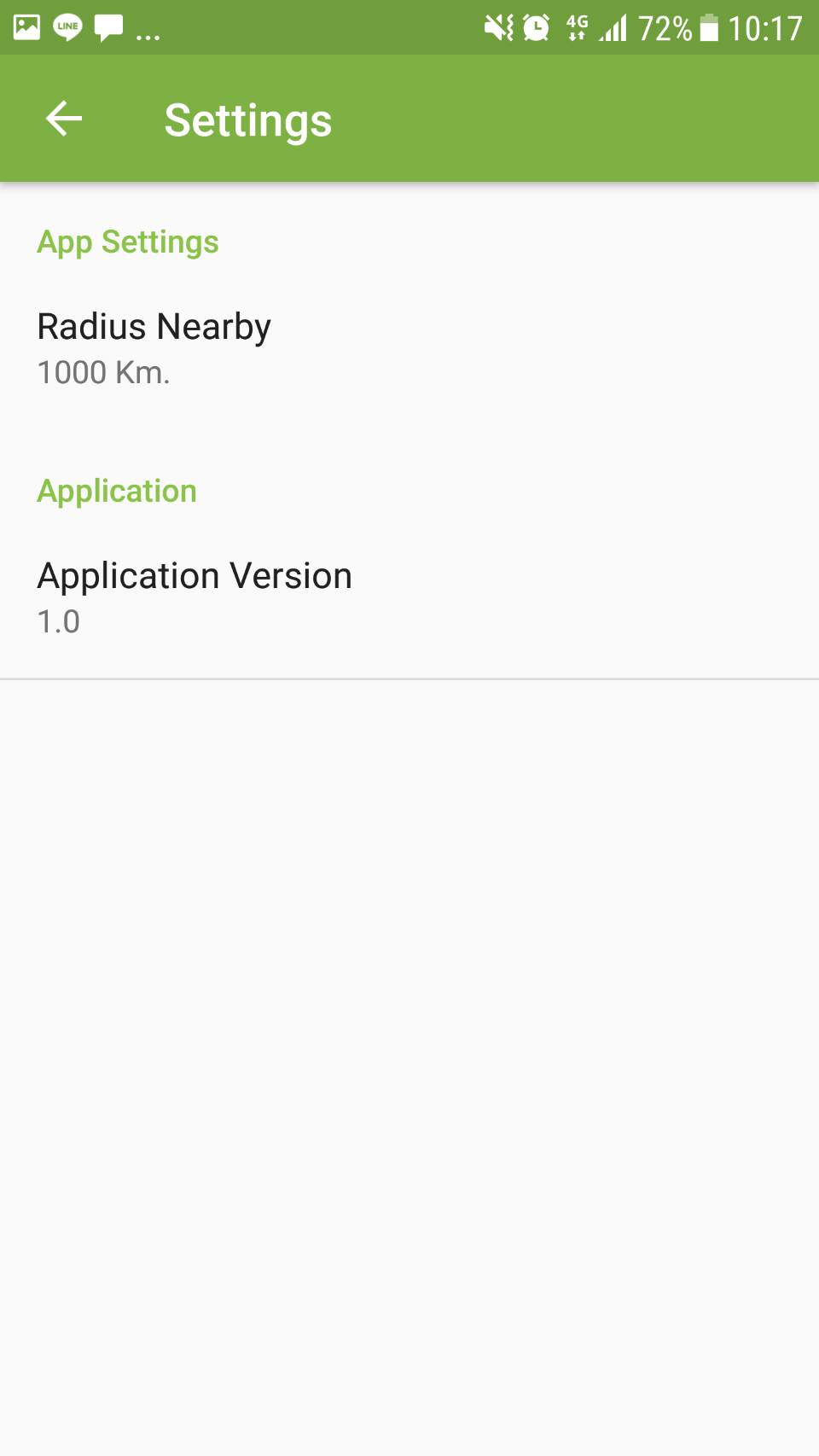


Figure 28. Menu Settings. [37]

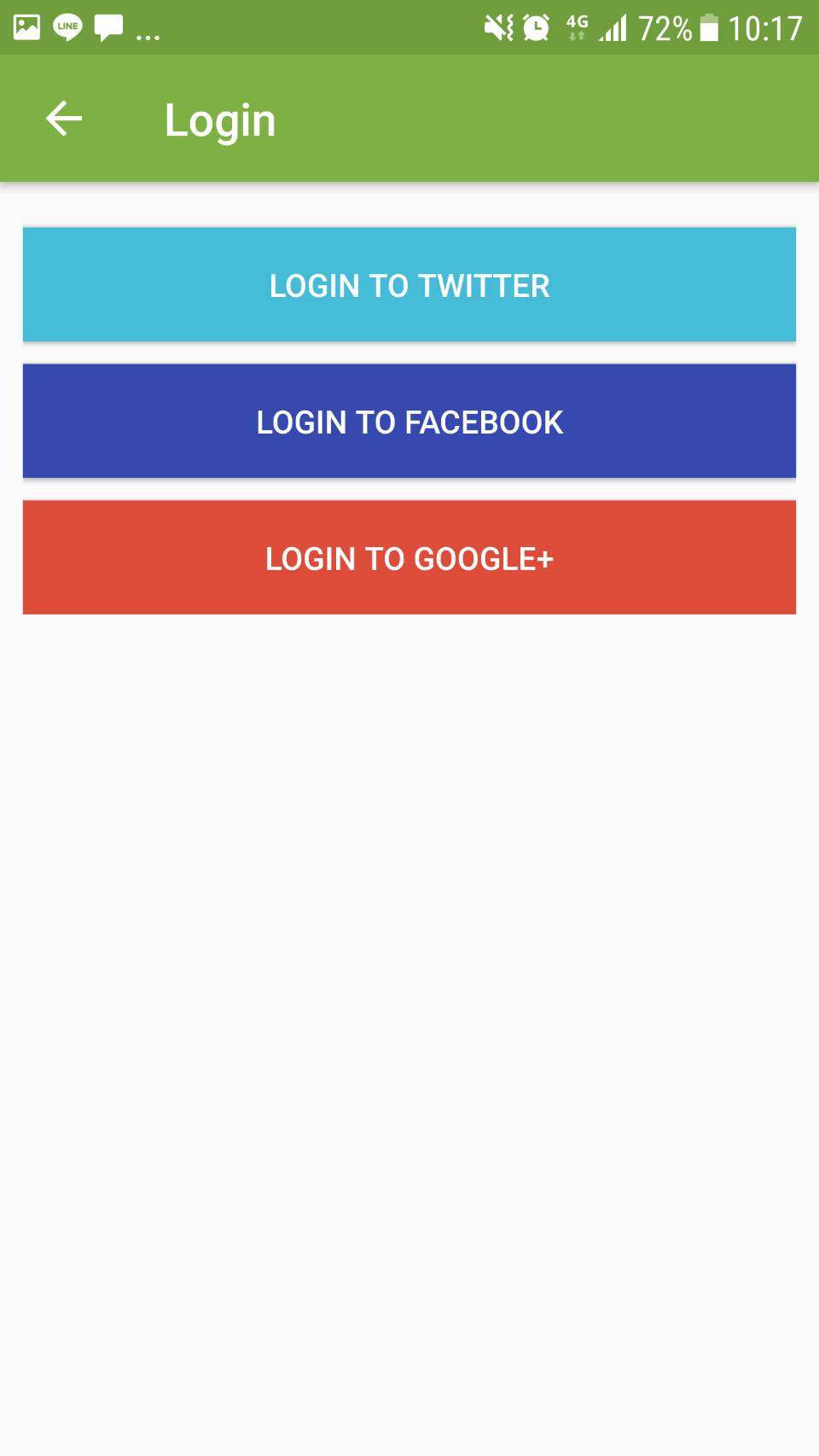


Figure 29. Menu Login. [37]

## MODIFIKASI YANG DILAKUKAN (SPLASH SCREEN, MENU, HOME, ABOUT US, DAN LOGIN) – Kevin Kurniawan

Untuk menyesuaikan terhadap fitur dari WhatsCamp, maka kami melakukan beberapa modifikasi dari *template* yang kami gunakan [37]. Modifikasi yang kami lakukan adalah sebagai berikut.

* *Splash Screen*.

Kami melakukan modifikasi dari tampilan *splash screen* untuk menyesuaikan dengan logo dan nama aplikasi kami, yaitu WhatsCamp. Sebelumnya, kami menentukan terlebih dahulu logo dari aplikasi kami. Kami menentukan logo apa yang cocok untuk menggambarkan aplikasi kami dengan tetap memperhatikan prinsip HCI secara keseluruhan, khususnya yang telah dibahas pada bagian *Literature Survey*: Hubungan Prinsip *Human-Computer Interaction* (HCI) dengan *User Interface* (UI) [5]. Maka dari itu, kami memutuskan untuk mengambil logo dengan gambar megafon (pengeras suara) karena dalam dunia nyata, megafon ini umumnya digunakan untuk memberikan informasi atau pemberitahuan kepada orang banyak dan aplikasi kami juga bertujuan untuk memberikan informasi kepada orang banyak mengenai acara apa saja yang akan diadakan. Dengan logo ini, kami telah menyesuaikan pilihan dengan memenuhi prinsip *Mental Models* [5].



Figure 30. Logo WhatsCamp. [38]

Logo WhatsCamp sendiri kami ambil dari calltoactionmedia.com [38] dan kami lakukan modifikasi untuk menambahkan tulisan WhatsCamp (nama aplikasi). Setelah itu, kami lakukan penggantian logo pada bagian *code* dengan langkah sebagai berikut.

1. Tambahkan logo untuk *splash screen* pada folder res/mipmap.

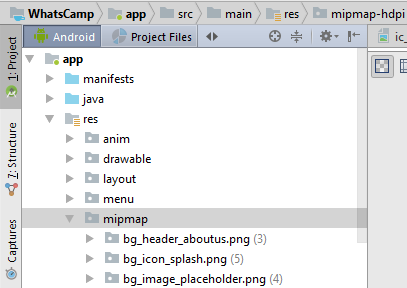


Figure 31. Folder mipmap.

1. Penambahan dilakukan dengan cara: klik kanan > New > Image Asset.

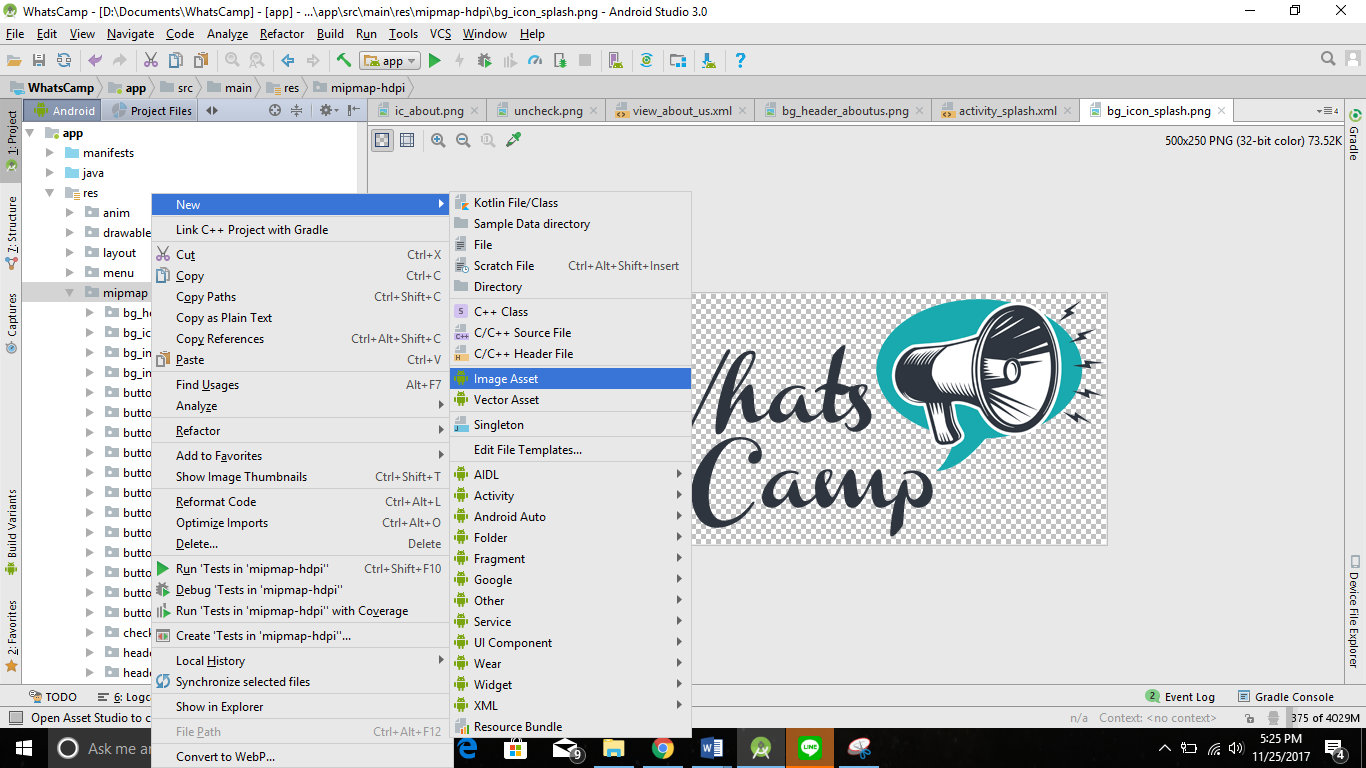


Figure 32. Penambahan Image Asset.

Penambahan dengan cara seperti ini diperlukan agar didapatkan gambar yang kita masukkan dengan resolusi yang berbeda-beda (mdpi, hdpi, xhdpi, xxhdpi, dan xxxhdpi). Penentuan resolusi gambar yang digunakan saat aplikasi dijalankan akan dilakukan secara otomatis dengan menyesuaikan resolusi layar yang digunakan untuk menampilkan aplikasi.

1. Hubungkan *image asset* yang telah dimasukkan dengan *layout* dari *splash screen* (activity\_splash.xml).

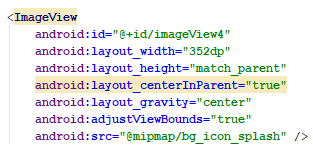


Figure 33. Menghubungkan Image Asset pada layout.

Kami juga melakukan penghapusan TextView yang ada karena logo kami sudah mengandung nama dari aplikasi.



Figure 34. Splash Screen awal. [37]



Figure 35. Splash Screen WhatsCamp.

* Menu.

Kami melakukan modifikasi dari tampilan *menu* untuk menyesuaikan fitur apa saja yang ada pada WhatsCamp. Kami melakukan penghapusan menu *Featured* dan *Latest News* dari *template* awal karena aplikasi WhatsCamp tidak memerlukan fitur tersebut. Penghapusan menu dilakukan dengan menghapus *item* dalam 2 (dua) buah *file* XML pada folder layout/menu (activity\_main\_drawer\_logged.xml dan activity\_main\_drawer\_not\_logged.xml) karena menu pada saat sebelum dan setelah melakukan *login* berbeda.

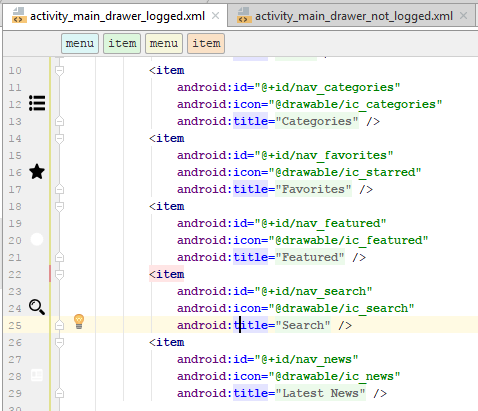


Figure 36. Item-item dari menu.

Kami juga melakukan modifikasi pada *icon-icon* *item* dari menu. Modifikasi dilakukan untuk menyesuaikan kembali *icon-icon* dari menu dengan prinsip HCI yang telah dibahas pada bagian *Literature Survey* sebelumnya. *Icon* kami sesuaikan agar memberikan gambaran yang sesuai dengan fungsi dari menu itu sendiri, sehingga memenuhi prinsip *affordance* [5].Semua *icon* baru yang akan kami terapkan pada pengubahan, kami dapatkan dari flaticon.com [39]. *Icon-icon* yang kami ubah adalah sebagai berikut.

Tabel 1 – Perubahan *Icon*

|  |  |  |  |
| --- | --- | --- | --- |
| Menu | Icon Lama | Icon Baru | Alasan |
| Home | [37] | [39] | Simbol rumah biasanya digunakan untuk merepresentasikan metode untuk kembali ke halaman awal. [40] |
| Categories | [37] | [39] | Simbol tumpukan dengan titik di samping kiri setiap tumpukkan melambangkan beberapa kategori yang umumnya terdapat gambar (titik) dan tulisan (setiap garis pada tumpukkan). |
| Favorites | [37] | [39] | Simbol bintang biasanya digunakan untuk merepresentasikan favorit. [40] |
| Search | [37] | [39] | Lingkaran dengan garis ke arah kanan bawah menyerupai kaca pembesar merupakan simbol yang biasanya digunakan untuk pencarian. [40] |
| About Us | [37] | [39] | Simbol huruf “i” di dalam lingkaran melambangkan informasi. Dalam hal ini, informasi yang dimaksud adalah informasi dari pengembang aplikasi. |
| Terms and Conditions | [37] | [39] | Simbol kertas dengan tulisan melambangkan kertas perjanjian. Perjanjian yang dimaksud dalam penggunaan *icon* ini adalah syarat dan ketentuan dalam penggunaan aplikasi. |
| Settings | [37] | [39] | Simbol roda gigi (*gear*) melambangkan mengindikasikan opsi konfigurasi atau pengaturan (*settings*). [40] |
| Login | [37] | [39] | Simbol pintu yang terbuka dengan panah mengarah ke dalam pintu melambangkan untuk masuk ke dalam sesuatu. Dalam hal ini, masuk yang dimaksud adalah *login*. |
| Profile | [37] | [39] | Simbol orang melambangkan pengguna aplikasi dan digunakan untuk menggambarkan profil pengguna dalam aplikasi. |
| My Events | [37] | [39] | Simbol megafon melambangkan logo dari WhatsCamp. Kami menggunakan ini pada bagian My Events karena fitur dalam menu ini merupakan fitur utama dalam aplikasi WhatsCamp. |

Berikut merupakan cara yang dilakukan untuk mengganti *icon* dari menu.

* + - 1. Download *icon* dengan *file extension* .psd.

1. Penambahan dilakukan dengan cara: klik kanan > New > Image Asset.

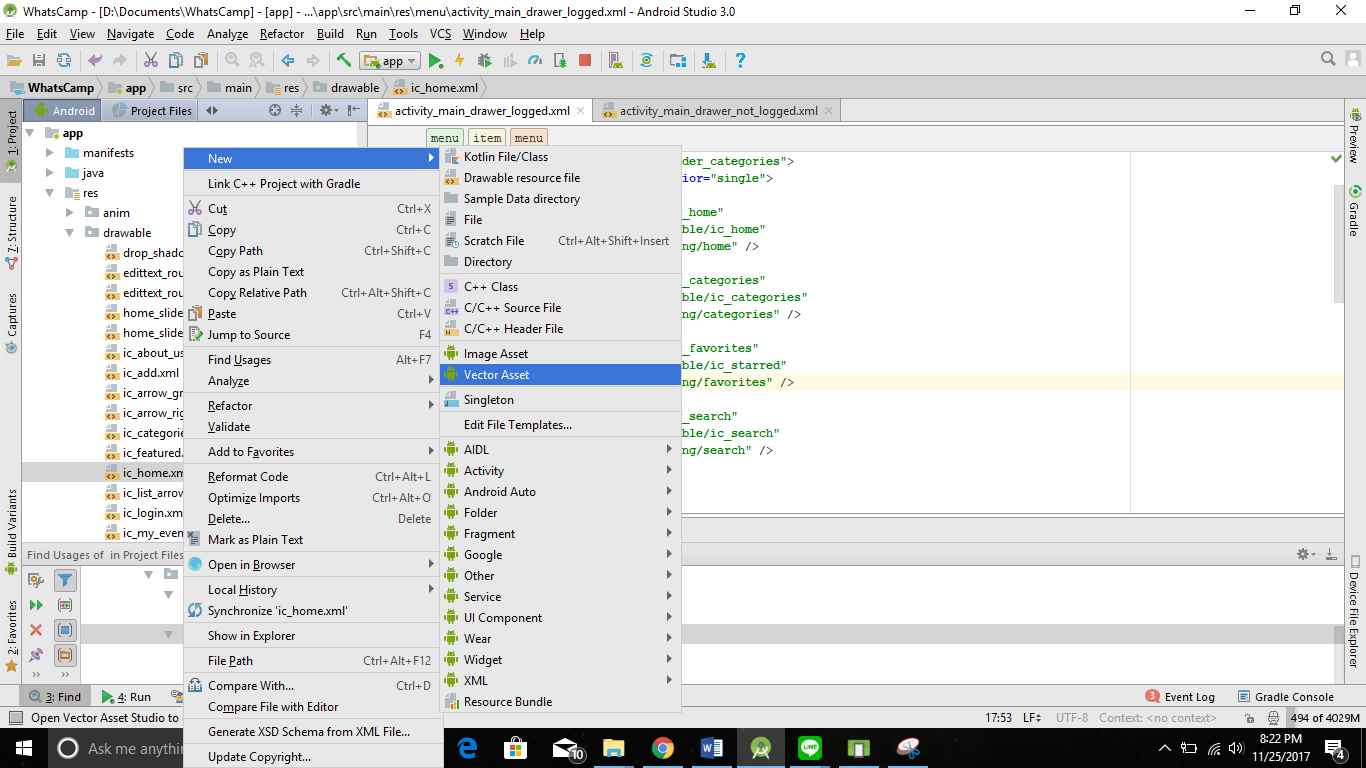


Figure 37. Penambahan Vector Asset.

Penambahan dilakukan harus melalui *vector asset* agar secara otomatis *icon* tersebut dapat berubah *extension* nya menjadi .xml setelah dimasukkan.



Figure 38. Contoh XML dari Vector Asset.

1. Hubungkan *vector asset* yang telah dimasukkan dengan XML dari menu tersebut (activity\_main\_drawer\_logged.xml dan activity\_main\_drawer\_not\_logged.xml).



Figure 39. Contoh menetapkan icon untuk menu pada XML.

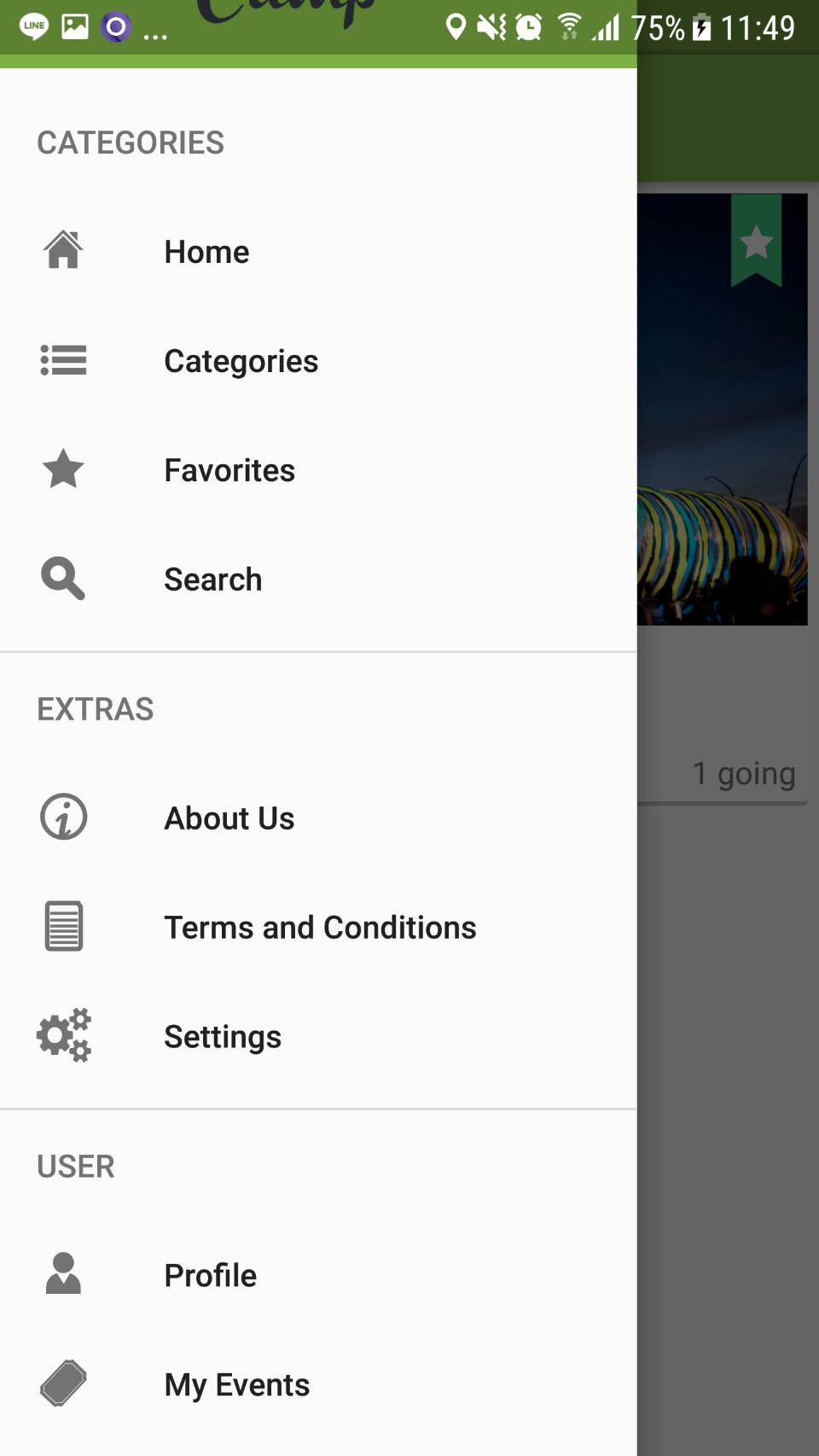
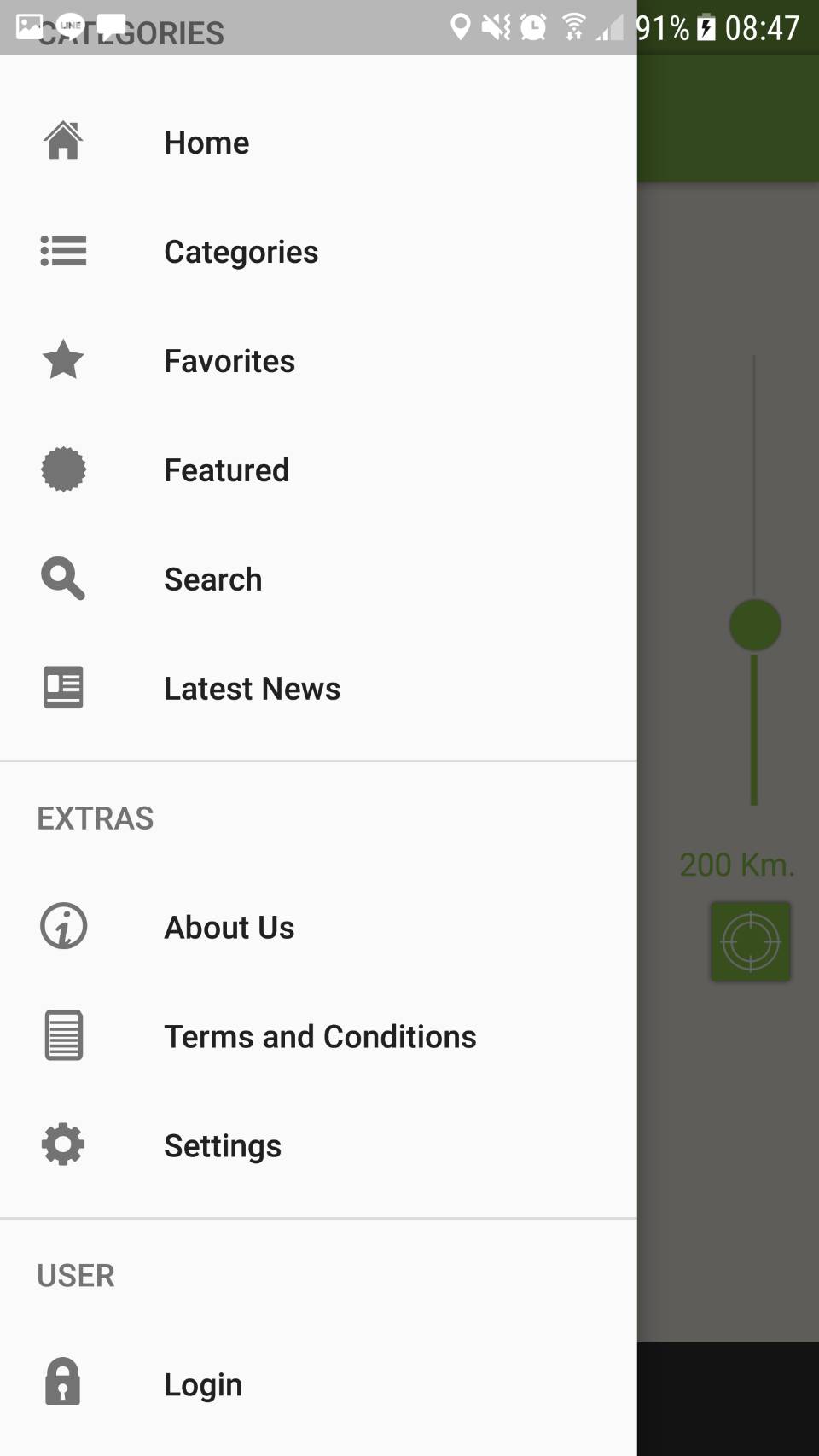


Figure 40. Menu awal. [37]

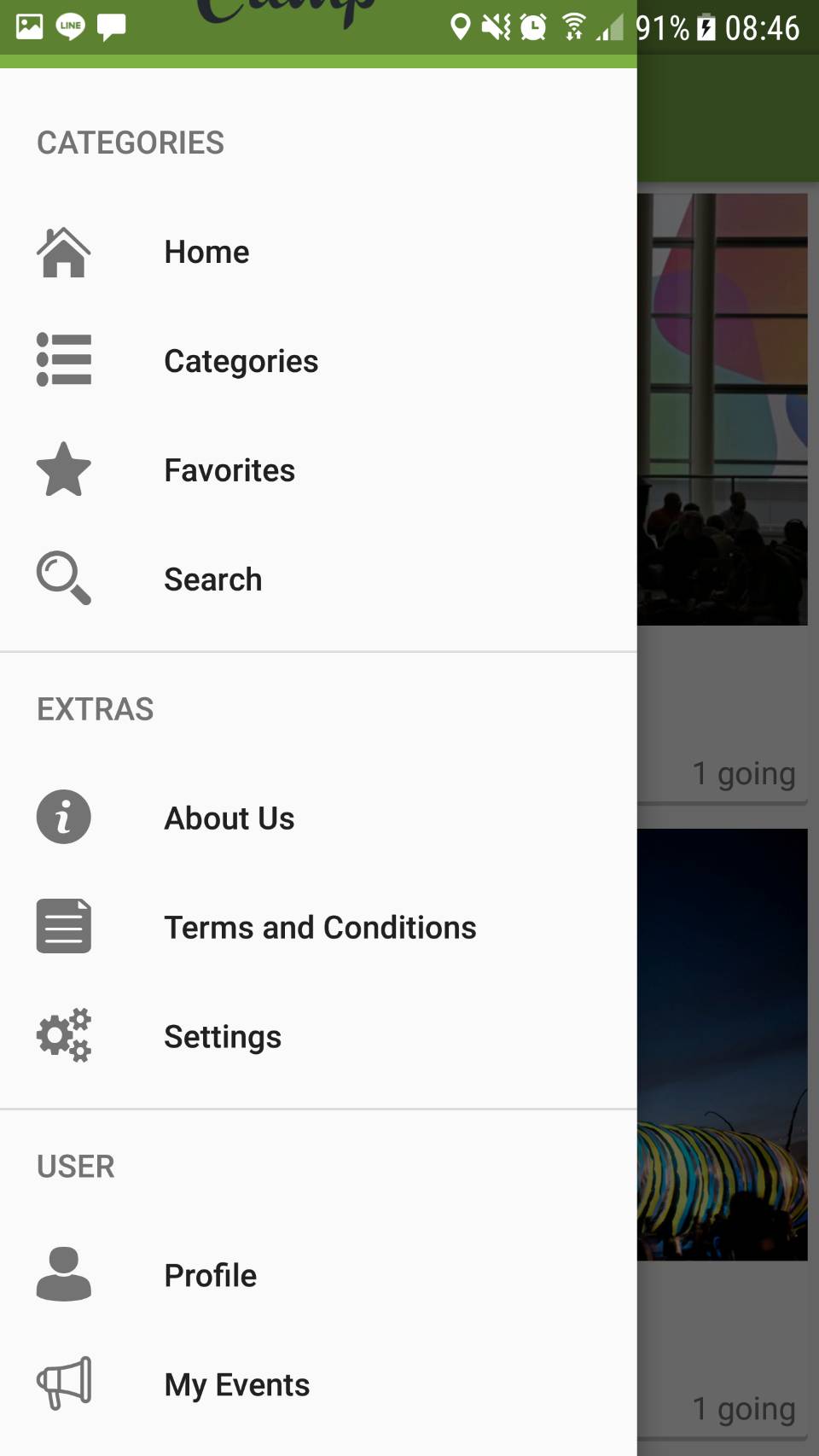
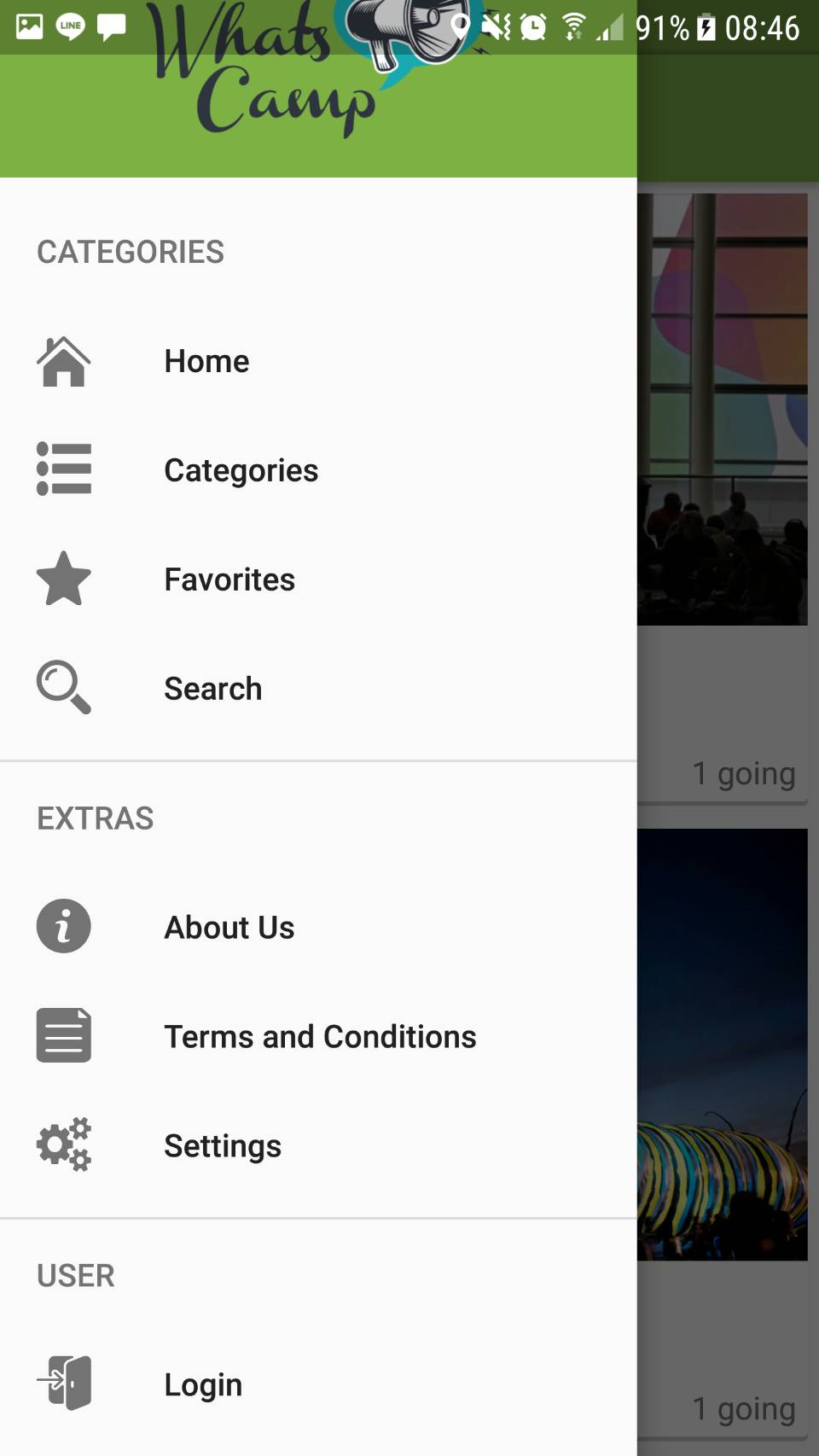


Figure 41. Menu modifikasi.

* Home.

Kami melakukan perubahan pada *layout* untuk menu Home (view\_home\_sliding.xml) dengan menyesuaikan rancangan dari aplikasi WhatsCamp. Maka dari itu, kami merubah komponen *layout* dari *template* awal untuk menu Home menjadi sebagai berikut.

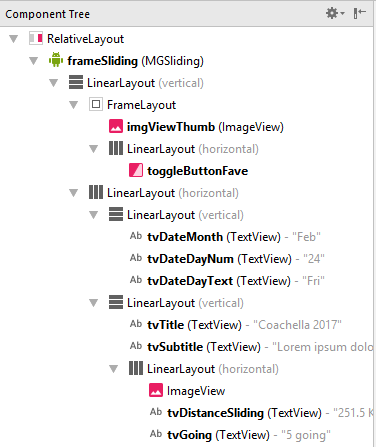


Figure 42. Component Tree untuk menu Home.

Komponen dari menu Home di-desain seperti ini untuk menampilkan keseluruhan *event* yang akan diselenggarakan.

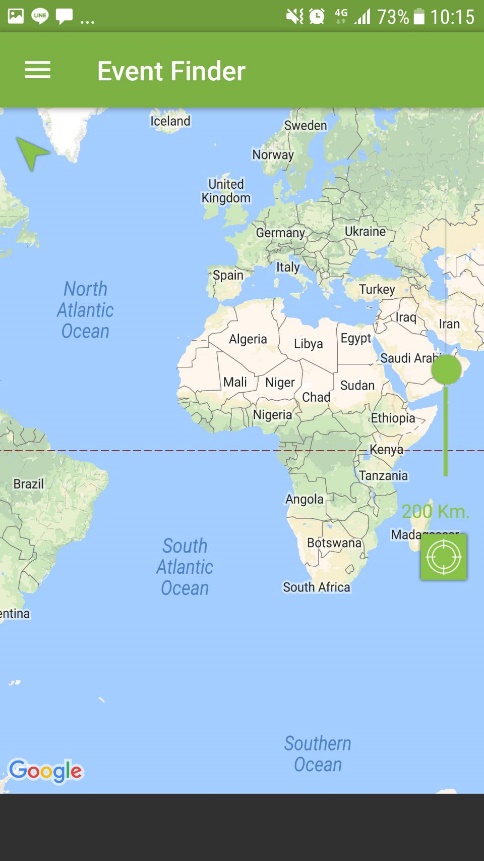


Figure 43. Isi Home awal. [37]

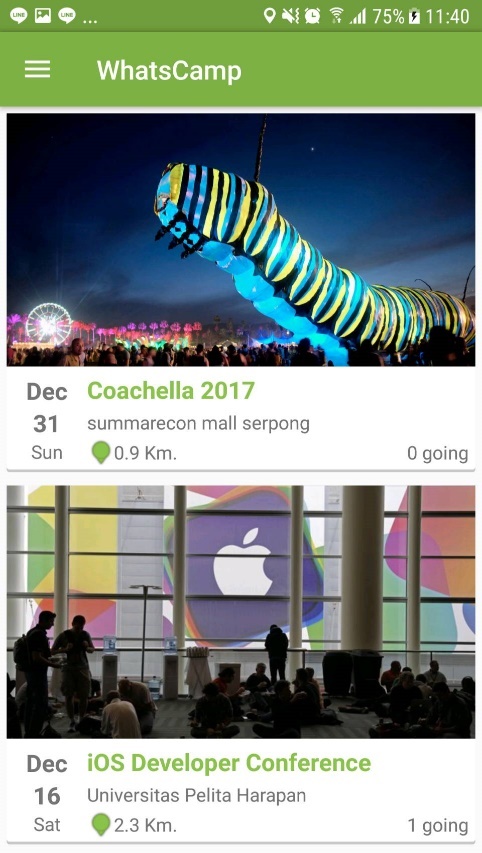


Figure 44. Isi Home yang telah dimodifikasi.

* About Us

Kami melakukan modifikasi pada isi menu About Us yang awalnya diletakkan gambar dari pengembang *template* menjadi gambar dari WhatsCamp. Sebelumnya, kami membuat gambar dari logo WhatsCamp yang diberi *background* dengan ukuran 640x500 piksel.



Figure 45. Logo WhatsCamp dengan background.

Selanjutnya, kami melakukan modifikasi pada *code* dengan langkah sebagai berikut.

1. Tambahkan gambar yang telah dibuat dalam folder res/mipmap. Penambahan dilakukan dengan cara: klik kanan > New > Image Asset.

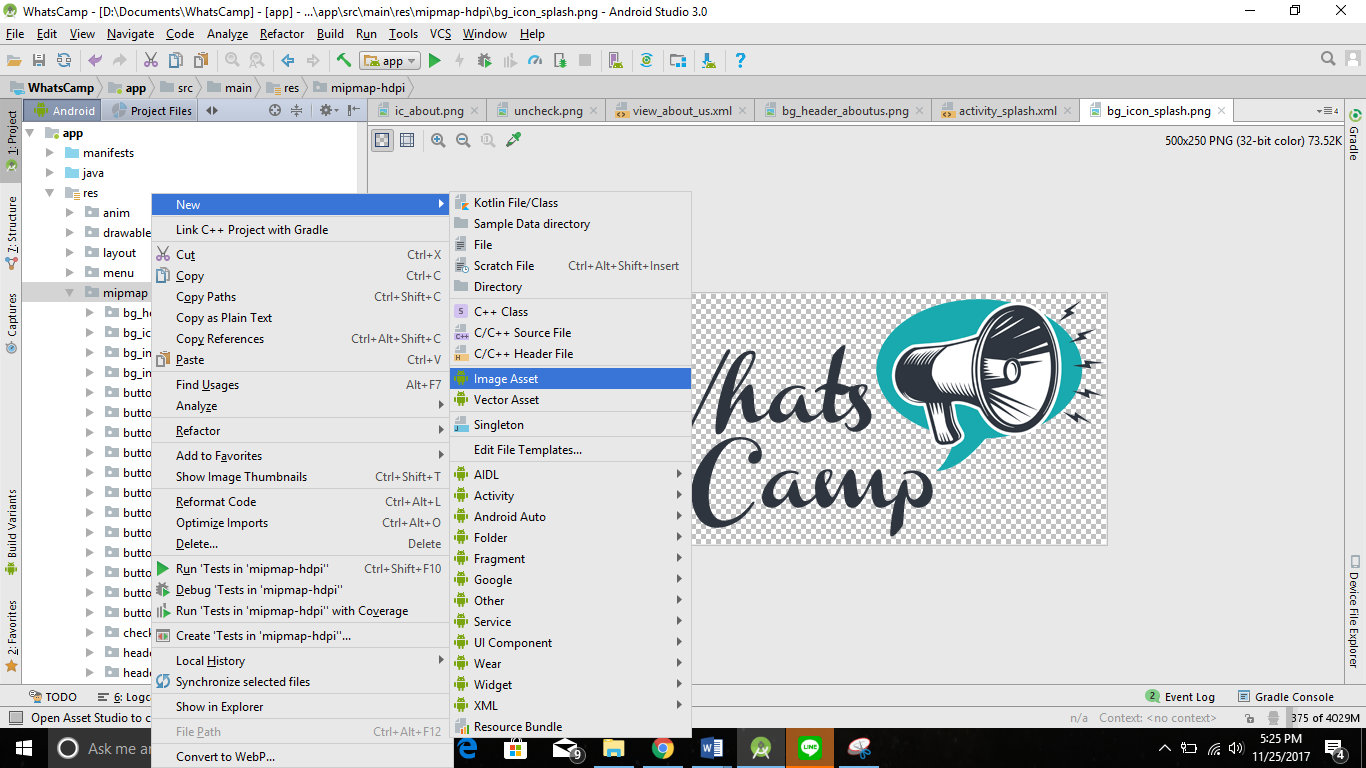


Figure 46. Penambahan Image Asset.

1. Lalu, hubungkan gambar yang telah ditambahkan dengan *layout* menu About Us (view\_about\_us.xml).

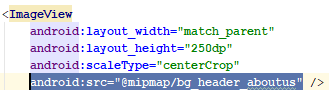


Figure 47. Menambahkan gambar pada layout.

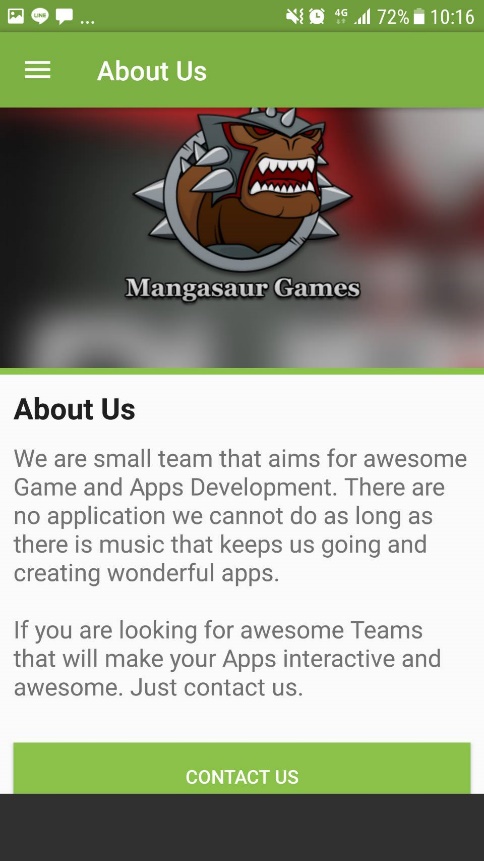


Figure 48. Isi menu About Us awal. [37]



Figure 49. Hasil modifikasi menu About Us.

* Login.

Dikarenakan *login* pada aplikasi WhatsCamp dirancang hanya melalui Facebook dan Google+, maka kami melakukan penghapusan terhadap tombol untuk *login* dengan Twitter yang ada di halaman Login pada *layout* awal. Penghapusan ini tombol ini dilakukan pada *file* view\_activity\_login.xml yang terdapat pada *folder* res/layout.



Figure 50. Tombol Twitter yang dihapus.

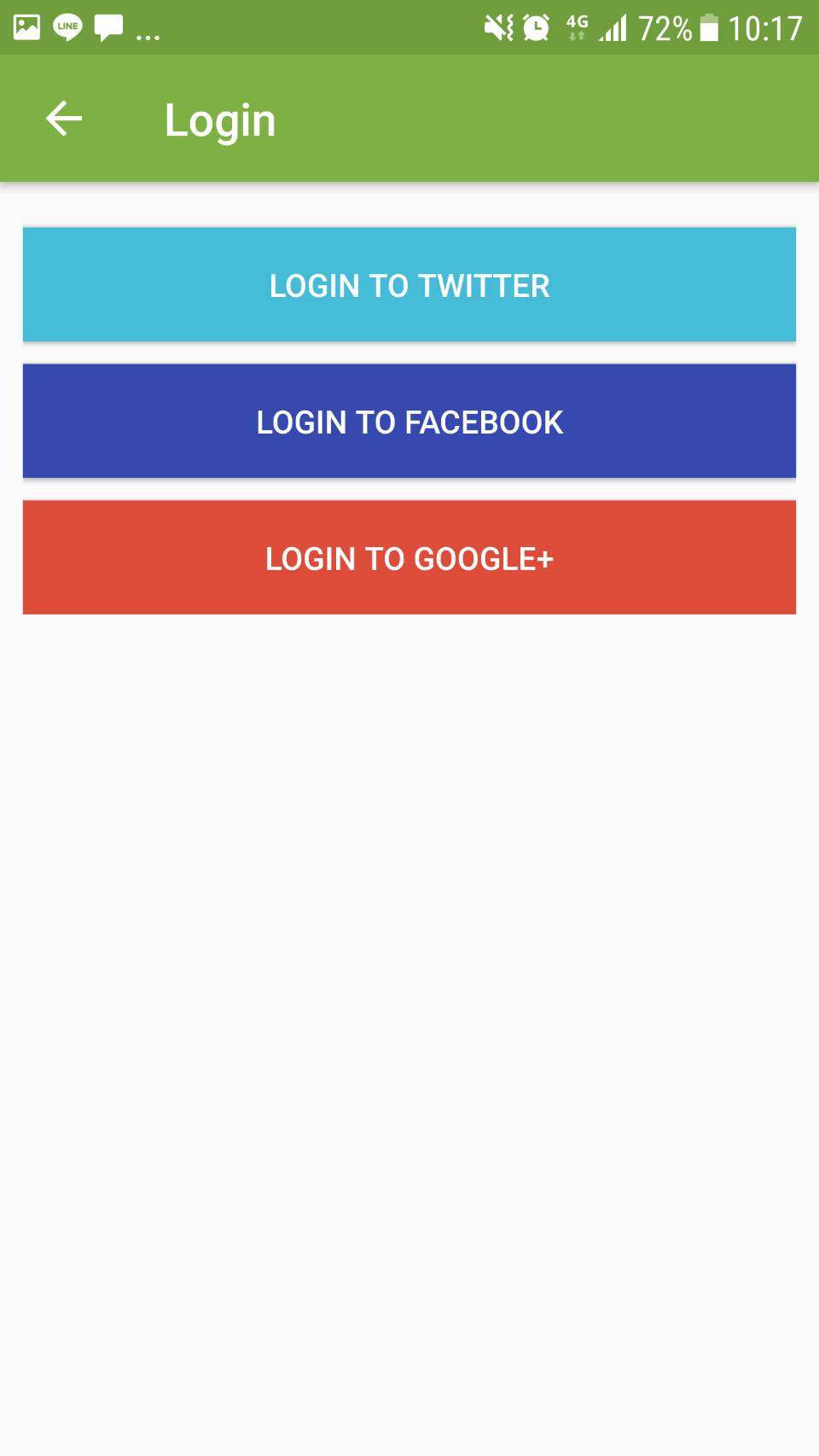


Figure 51. Menu Login awal. [37]

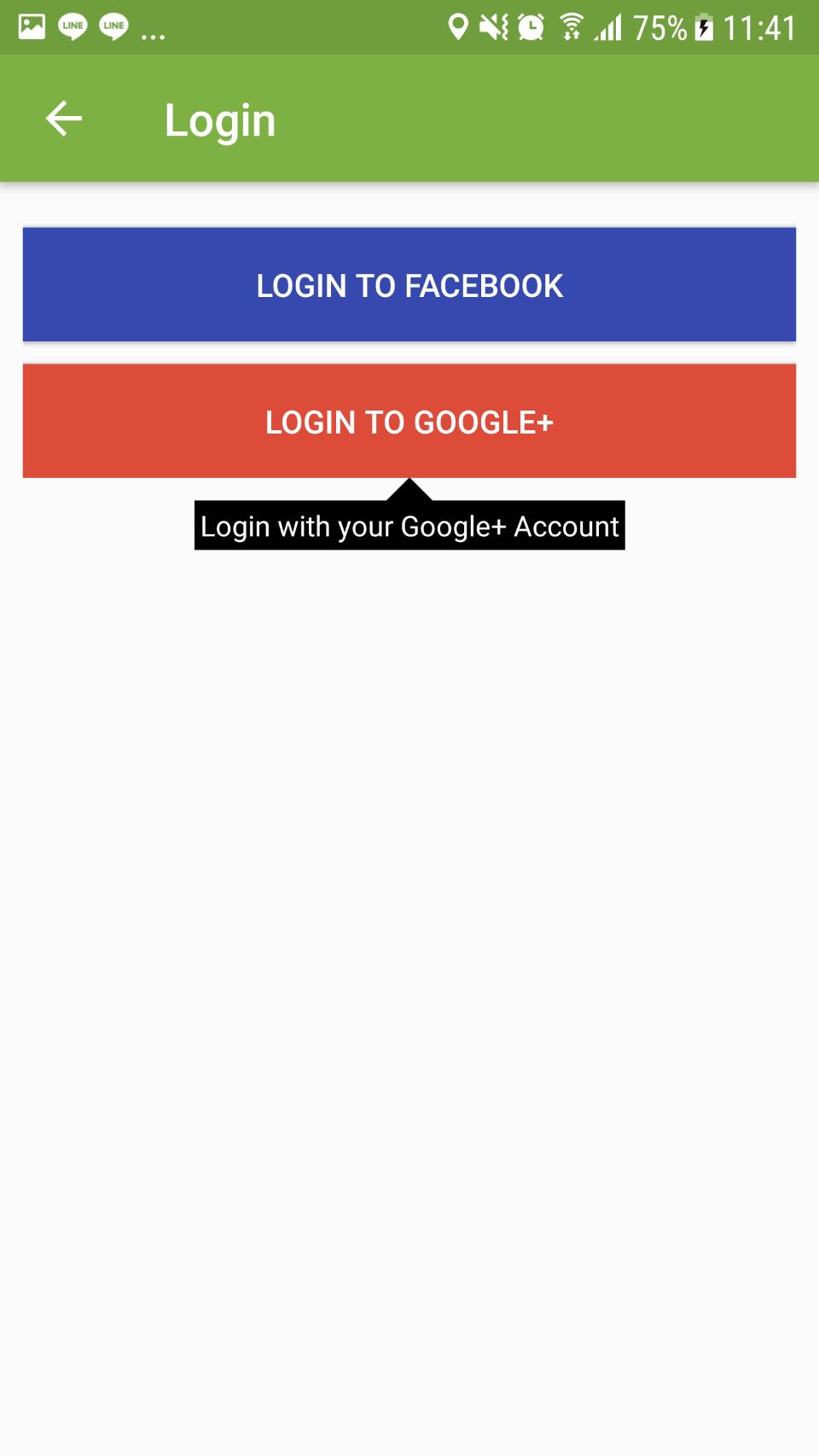


Figure 52. Hasil modifikasi menu Login.

## PROTOTIPE – Kevin Kurniawan

**Eventbrite**

Kami mengambil prototipe dari aplikasi Eventbrite [41], dimana aplikasi ini memiliki fungsi yang mirip dengan WhatsCamp, yaitu untuk mencari *event* yang akan diadakan. Namun, perbedaannya adalah WhatsCamp secara spesifik diperuntukkan bagi *event* yang akan diadakan oleh kampus, sedangkan Eventbrite [41] diperuntukkan bagi *event* secara umum yang tidak dibatasi wilayahnya. Akun dari pengguna Eventbrite [41] dikelola sendiri oleh Eventbrite [41] (*sign-up* dan *sign-in*), sedangkan akun dari pengguna WhatsCamp terintegrasi dengan Facebook dan Google+. Fitur dari Eventbrite [41] yang juga dimiliki oleh WhatsCamp adalah *view event*, *search event*, *going to event*, dan *favorite event*.

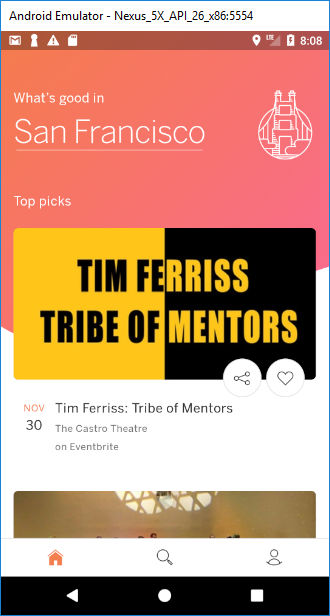


Figure 53. Menu Home Eventbrite. [41]

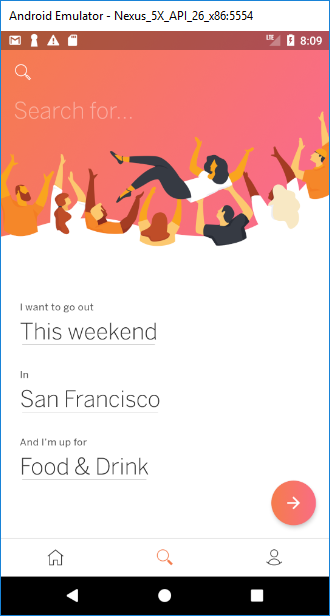


Figure 54. Fitur search Eventbrite. [41]

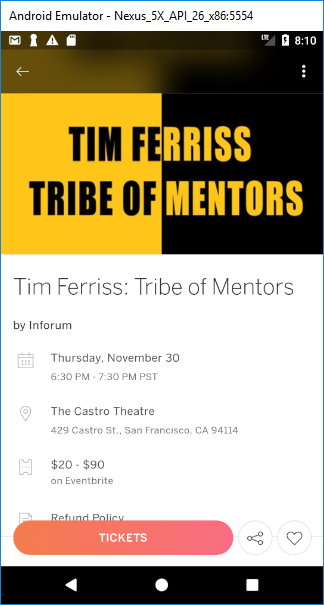


Figure 55. View Event Eventbrite. [41]

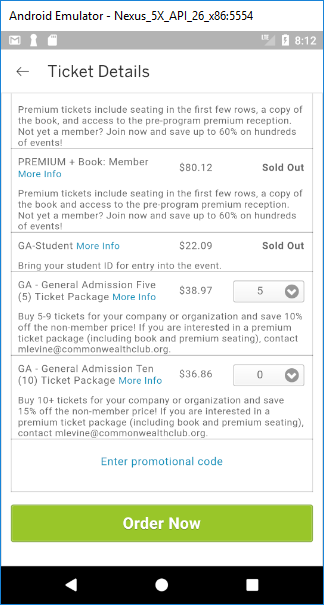


Figure 56. Fitur pembelian tiket Eventbrite. [41]

Kelebihan Eventbrite [41]:

* Dapat melakukan pembelian tiket secara langsung melalui aplikasi.
* Terdapat fitur pencarian *event* melalui waktu, wilayah, dan/atau minat.
* Terdapat *welcome screen* di antara *splash screen* dan menu utama.

Kelemahan Eventbrite [41]:

* *Event* tidak dapat ditambahkan oleh pengguna. Penambahan *event* hanya dapat dilakukan oleh pihak *event* yang bersangkutan.

## KREATIFITAS DAN INOVASI – Kevin Kurniawan

Setelah kami mencoba menggunakan aplikasi Eventbrite, kami menyimpulkan kelebihan dari WhatsCamp adalah sebagai berikut.

* *Login* menggunakan API dari Facebook dan Google+. Data dari pengguna diambil secara otomatis melalui akun Facebook atau Google+ yang digunakan untuk *login*.

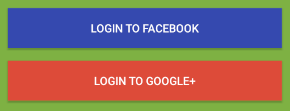


Figure 57. Login dengan Facebook atau Google+.

* Tidak hanya *admin* yang dapat menambahkan *event*. Semua pengguna dapat melakukan penambahan *event* di dalam aplikasi.



Figure 58. User dapat menambahkan event. (1)

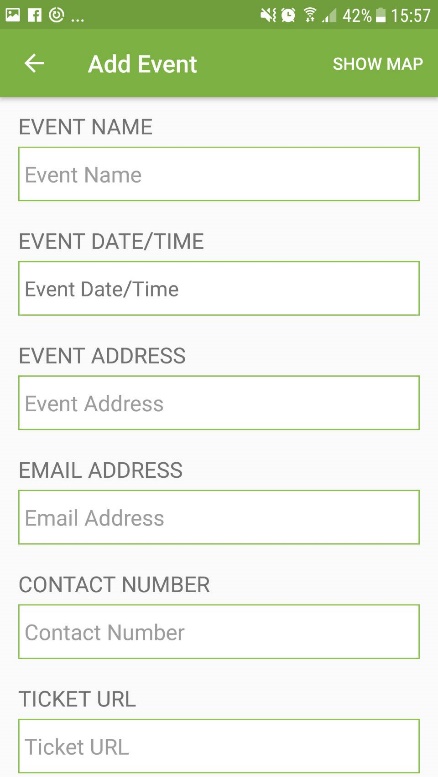


Figure 59. User dapat menambahkan event. (2)

* *User* dapat melakukan *Join Event* dan dapat diketahui oleh *user* lain yang akan mendatangi *event* yang sama.

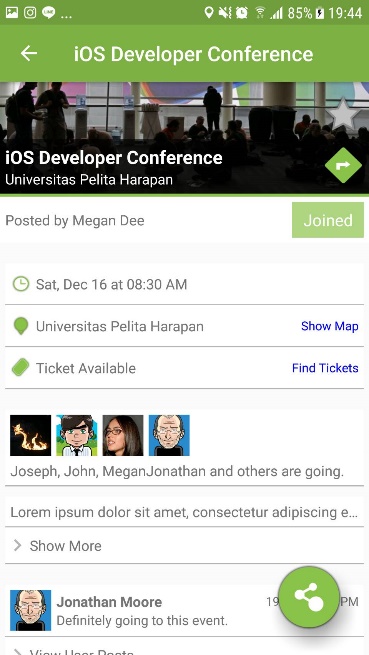


Figure 60. Join Event.

* *User* dapat melakukan kontak langsung (melalui telepon) dengan *contact person* yang tertera di dalam deskripsi *event.*



Figure 61. Telepon contact person.

## REDESIGN – Kevin Kurniawan

Setelah kami membandingkan WhatsCamp dengan Eventbrite [41], kami memutuskan untuk menambahkan *welcome screen* pada WhatsCamp. *Welcome screen* yang ada pada aplikasi Eventbrite [41] mencakup pemilihan wilayah, pemilihan minat, dan *login*. Kami memutuskan untuk menambahkan *welcome screen* hanya untuk *login* saja karena WhatsCamp tidak memiliki fitur pemilihan atau pencarian *event* berdasarkan wilayah dan minat. Kami merasa penambahan *welcome screen* pada aplikasi kami diperlukan agar pengguna mengetahui bahwa untuk dapat menggunakan fitur dari aplikasi secara maksimal, maka pengguna harus melakukan *login* terlebih dahulu.

Berikut merupakan cara penambahan *welcome screen* pada aplikasi WhatsCamp.

1. Menambahkan activity baru yang meliputi WelcomeScreenActiviy.java dan activity\_welcome\_screen.xml
2. Mengatur desain dari *welcome* *screen* pada bagian activity\_welcome\_screen.xml

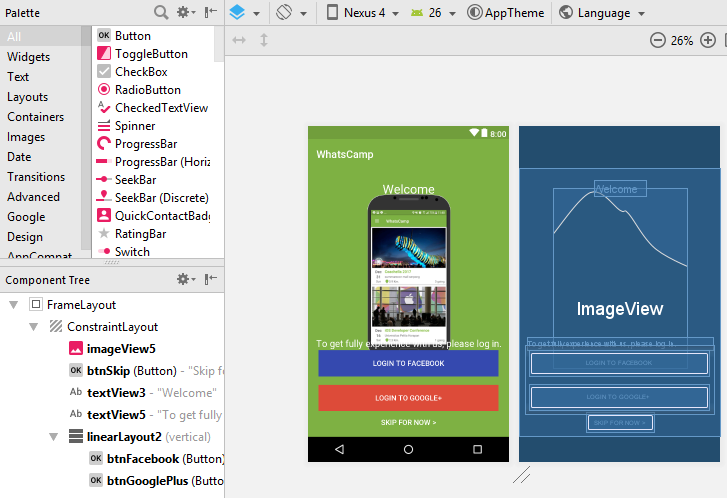


Figure 62. activity\_welcome\_screen.xml

1. Mengatur Intent dari *splash screen* menuju *welcome screen* pada SplashActivity.java.

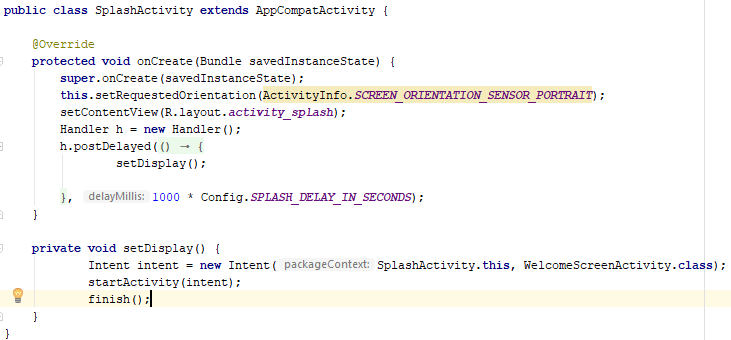


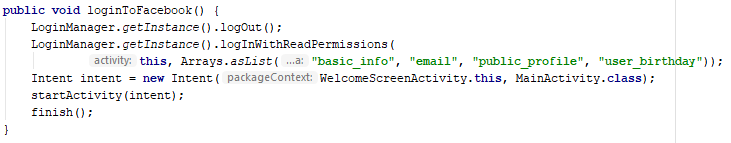
Figure 63. SplashActivity.java

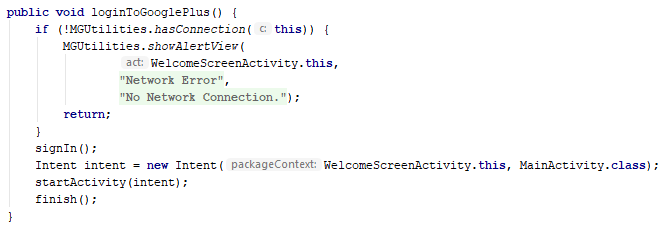
1. Menambahkan fitur login pada tombol login.



Figure 64. WelcomeScreenActivity.java

1. Menambahkan Intent dari tombol login dan tombol skip menuju MainActivity.java (halaman utama aplikasi).





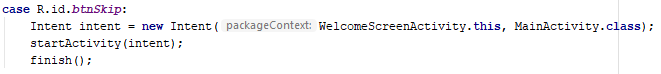


Figure 65. Intent yang ditambahkan.

Hasil dari *welcome screen* yang kami tambahkan adalah sebagai berikut.

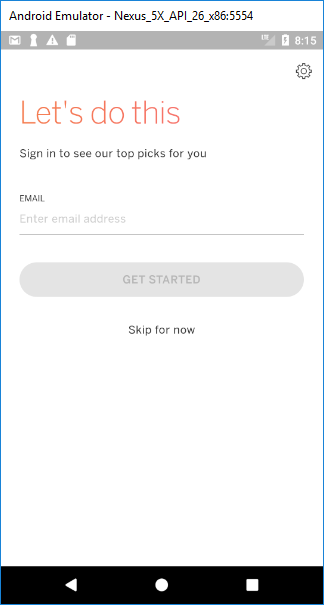


Figure 66. Welcome screen Eventbrite. [41]

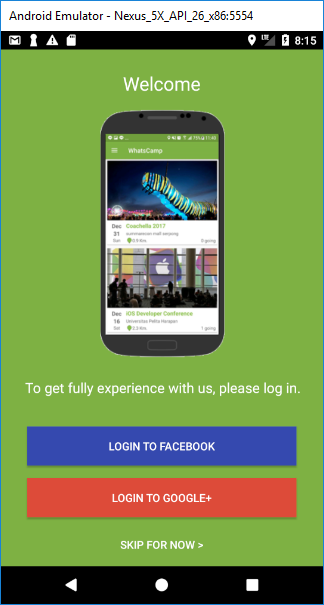


Figure 67. Welcome screen WhatsCamp.

## *LIMITATION CONSIDERATION –* Joseph

Pada aplikasi “Whatscamp” yang kami buat, ada beberapa batasan yang digunakan, yaitu

* 1. Penggunaan MAMP

Aplikasi kami menggunakanan MAMP sebagai dasar untuk melakukan hosting database kami karena MAMP secara langsung mengkombinasikan MAC, Apache, MySQL dan PHP menjadi satu. MAMP juga merupakan *platform* yang *open-source,* berfokus pada Windows dan MAC*,* mampu mensetup lingkungan dan mempunyai proses yang mudah*.* PHP yang digunakan dapat menggunakan *object-oriented scripting language* [42][43]*.*

* 1. Login dibatasi Facebook dan Google+

Penggunaan Facebook dan Google+ dapat membatasi dalam menggunakan *email* fiktif atau *email* yang terdaftar pada kedua *social media*. Karena login disupport oleh API, maka login secara langsung di *direct* ke Facebook maupun Google+ sehingga apabila tidak terdaftar, maka tidak dapat dilakukan login karena ada *direct* ke Google+ atau Facebook.

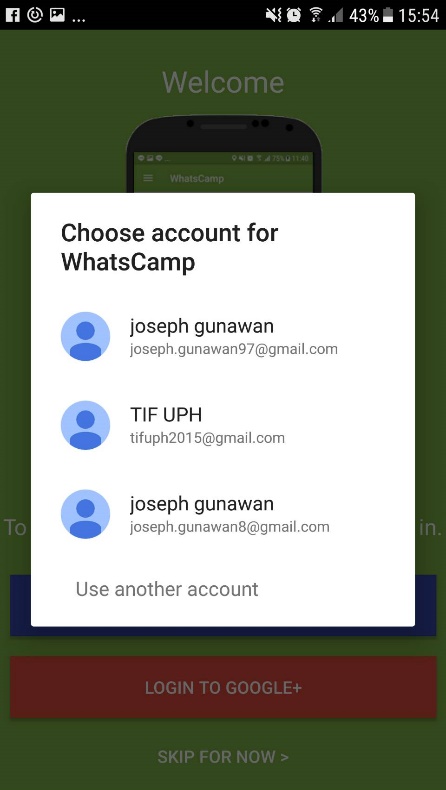
**

Figure 68. Login dengan Google+

# BAB IV

# *OTHER WORKING PROGRAM/EXCELLENCE, COMPLEXITY OF CODING* – Jessica Sean

Pada proyek ini, terdapat kompleksitas dari code. Kita menggunakan bahasa pemogramman Java untuk User Interface (UI), desain dan tampilan yang dimunculkan dalam activity. Pada back-end, kamu menggunakan php untuk melakukan koneksi terhadap database. Kompleksitas dari code dapat dilihat melalui hubungan Android Studio (Java) dengan file php. Dalam project whatscamp ini, kami menggunakan MySQL sebagai database management system, sehingga untuk melakukan update ataupun mengambil data dari database, dibutuhkan query. Query itu sendiri disimpan dan dipanggil dalam fungsi-fungsi yang terdapat pada file php, yang terletak pada C:/MAMP/htdocs/whatscamp.

File php yang digunakan dalam proyek WhatsCamp ini dikategorisasikan ke dalam empat folder besar. Terdapat juga file header\_rest.php yang digunakan untuk mengkoneksi satu class dengan file model dan controller. Folder pertama dari terdiri dari empat file php sebagai berikut:

1. Config.php: untuk melakukan konfigurasi database dan inisiasi variable yang akan digunakan untuk pengaksesan database.

const DB\_HOST = "localhost";

const DB\_USER = "root";

const DB\_PASSWORD = "root";

const DB\_DATABASE = "db\_whatscamp";

const ROOT\_URL = "http://localhost/whatscamp/";

const IMAGE\_UPLOAD\_DIR = "upload\_pic";

const JSON\_FILE = "rest/data.php";

const NO\_OF\_ITEMS\_PER\_PAGE = 10;

const MAP\_DEFAULT\_LATITUDE = 37.770185;

const MAP\_DEFAULT\_LONGITUDE = -122.419061;

const MAP\_DEFAULT\_ZOOM\_LEVEL = 18;

const API\_KEY = "AIzaSyAx7bTQGYXrbjnrhDwcseM9JlbfcKo32FA";

1. DB\_Connect.php: terdiri dari class yang digunakan utnuk mengkoneksi database yang terdiri dari fungsi connect().
2. Extras.php: File php ini terdiri dari class Extras yang digunakan untuk melakukan query enkripsi dan dekripsi, untuk menghilangkan prefix http dan encrypt serta decrypt params.
3. Globals.php: Fungsi dari php ini adalah untuk mendefinisikan string DEBUG untuk menetapkan *show debugging* sebagai *true* dan variable KEY\_SALT yang membuat file-file tetap terproteksi.

Folder kedua merupakan folder controller yang mencakup beberapa php file yang terdiri dari fungsi-fungsi. Fungsi yang terdapat pada class akan digunakan untuk menjalankan query. Contohnya dapat dilihat pada fungsi getAttendeeByEventIdAndUserId($event\_id, $user\_id) pada ControllerAttendee.php yang digunakan untuk mengambil entry dengan parameter event id dan user id. Berikut merupakan fungsi tersebut:

public function getAttendeeByEventIdAndUserId($event\_id, $user\_id)

{

$stmt = $this->pdo->prepare('SELECT \* FROM tbl\_whatscamp\_attendees WHERE event\_id = :event\_id AND user\_id = :user\_id');

$stmt->execute( array('event\_id' => $event\_id, 'user\_id' => $user\_id) );

$array = array();

foreach ($stmt as $row)

{

$itm = new Attendee();

$itm->event\_id = $row['event\_id'];

$itm->is\_going = $row['is\_going'];

$itm->attendee\_id = $row['attendee\_id'];

$itm->is\_going = $row['is\_going'];

$itm->user\_id = $row['user\_id'];

$itm->created\_at = $row['created\_at'];

$itm->updated\_at = $row['updated\_at'];

$itm->is\_deleted = $row['is\_deleted'];

return $itm;

}

return null;

}

Berikut merupakan fungsi dari file php yang terdapat pada file Controller:

1. ControllerAttendee.php : untuk memodifikasi data pada tabel tbl\_whatscamp\_attendees.
2. ControllerCategory.php : untuk memodifikasi data pada tabel tbl\_whatscamp\_category.
3. ControllerEvent.php : untuk memodifikasi data pada tabel tbl\_whatscamp\_events table.
4. ControllerEventCategory.php : untuk memodifikasi data pada tabel tbl\_whatscamp\_event\_categories.
5. ControllerPost.php :untuk memodifikasi data pada tabel tbl\_whatscamp\_posts.
6. ControllerUser.php : untuk memodifikasi data pada tabel tbl\_whatscamp\_users.
7. ControllerRest.php: terdiri dari beberapa fungsi yang digunakan untuk mendapat event, kategori event, menghitung total post dan fungsi lainnya yang berhubungan dengan fitur pencarian.

Folder ketiga adalah folder model yang terdiri dari beberapa file yang merupakan abstract class atau object. Hal ini juga merunjuk kepada alasan mengapa kita menggunakan MAMP sebagai server. Kami telah mencoba untuk memindahkan project kami ke XAMPP, namun aplikasi dan database tidak berhasil dikoneksikan karena XAMPP tidak mendukung fitur object-oriented class.

Folder terakhir merupakan folder rest yang terdiri dari beberapa file php untuk menjalankan query dengan fungsi tertentu. Sebagai contoh, delete\_event.php digunakan untuk menghapis event dari database. Berikut merupakan code yang tertulis dalam file:

<?php

require '../header\_rest.php';

$controllerRest = new ControllerRest();

$controllerEvent = new ControllerEvent();

$controllerUser = new ControllerUser();

$event\_id = 0;

if( !empty($\_POST['event\_id']) )

$event\_id = $\_POST['event\_id'];

$user\_id = 0;

if( !empty($\_POST['user\_id']) )

$user\_id = $\_POST['user\_id'];

$login\_hash = "";

if( !empty($\_POST['login\_hash']) )

$login\_hash = $\_POST['login\_hash'];

$api\_key = "";

if(!empty($\_POST['api\_key']))

$api\_key = $\_POST['api\_key'];

if(!$controllerUser->isUserIdExistAndHash($user\_id, $login\_hash) || Constants::API\_KEY != $api\_key) {

$jsonArray = array();

$jsonArray['status'] = array('status\_code' => "3", 'status\_text' => "Invalid Access. Please relogin.");

echo json\_encode($jsonArray);

}

else if($event\_id > 0) {

$controllerEvent->deleteEvent($event\_id, 1);

$jsonArray = array();

$jsonArray['status'] = array('status\_code' => "-1", 'status\_text' => "Success.");

$jsonArray['event'] = array('event\_id' => $event\_id);

echo json\_encode($jsonArray);

}

else {

$jsonArray = array();

$jsonArray['status'] = array('status\_code' => "3", 'status\_text' => "Invalid Access. Please relogin.");

echo json\_encode($jsonArray);

}

?>

Menurut pendapat kami, menggunakan file php untuk mengakses database dapat dikonsiderasikan sebagai kompleksitas code yang tinggi. Hal ini dikarenakan pada umumnya digunakan class DBOpenHelper untuk melakukan akses dan mengambil data dari database SQLite. Dalam project ini, kami menggunakan php untuk mengatur data kami dan melakukan operasi serta menyimpan query dari database.

# REFERENSI

[1] K. Danani, P. Bisht, R. Rautela, and I. Bhattacharya, “User Interface Design for Web And Mobile Devices,” *Int. Res. J. Eng. Technol.*, pp. 2395–56, 2017.

[2] R. A. Soni and A. Professor, “A STUDY PAPER ON ANDROID UI,” *Int. J. Enterp. Comput. Bus. Syst. ISSN (Online*, vol. 2, no. 1, pp. 2230–8849, 2013.

[3] C. Abras, D. Maloney-Krichmar, and J. Preece, “User-Centered Design,” 2004.

[4] I. Begum, “Hci and Its Effective Use in Design and Development of Good User Interface,” vol. 3, no. 3, pp. 176–180, 2014.

[5] T. O’Hear and Y. Boudjenane, *Human-Computer Interaction. Interacting in Various Application Domains*, vol. 5613, no. PART 4. 2009.

[6] M. Helander, *A Guide to Human Factors and Ergonomics*, vol. 51, no. 6. 2006.

[7] Linda Lund, “Aesthetics in User Interface Design: The Influence on Users’ Preference, Decoding and Learning,” Blekinge Institute of Technology, 2015.

[8] Umakant Mishra, *Improving Graphical User Interface Using TRIZ - Umakant Mishra - Google Buku*. 2009.

[9] C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*. Cengage Learning, 2016.

[10] R. Chopra, *Database management systems (DBMS) : a practical approach : concepts, principles, case studies, experiments*. S. Chand, 2010.

[11] T. Brown, “How to Choose a DBMS,” *Focus Commun. Proc. 5th Annu. Connect. Technol. Symp.*, p. 70, 1995.

[12] J. Stephens and C. Russell, *Beginning MySQL database design and optimization : from novice to professional*. Apress, 2004.

[13] S. K. Aditya, V. K. Karn, and P. Mohanta, *Android SQLite essentials : develop android applications with one of the most widely used database engines, SQLite*. .

[14] GeeksforGeeks, “ACID Properties in DBMS - GeeksforGeeks.” [Online]. Available: http://www.geeksforgeeks.org/acid-properties-in-dbms/. [Accessed: 02-Dec-2017].

[15] C. Singh, “ACID properties in DBMS.” [Online]. Available: https://beginnersbook.com/2015/04/acid-properties-in-dbms/. [Accessed: 02-Dec-2017].

[16] R. L. Boring, “Human-Computer Interaction as Cognitive Science,” *Proc. Hum. Factors Ergon. Soc. 46yh Annu. Meet.*, vol. 46, no. 21, pp. 1767–1771, Sep. 2002.

[17] P. R. Martin *et al.*, *IAAP Handbook of Applied Psychology*. Oxford, UK: Wiley-Blackwell, 2011.

[18] B. M. Velichkovsky, “Modularity of Cognitive Organization: Why It Is so Appealing and Why It Is Wrong,” in *Modularity: Understanding the Development and Evolution of Natural Complex Systems*, no. 1, 2005, pp. 353–381.

[19] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*. 2000.

[20] A. D. Swain and H. E. Guttmann, *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications*. 1983.

[21] J. Rasmussen, *Skills Rules and Knowledge, Other Distinctions in Human Performance Models*, vol. 13, no. 3. 1983.

[22] E. Hollnagel and D. Woods, “Joint cognitive systems: Foundations of cognitive systems engineering,” 2005.

[23] P. C. Cacciabue, *Guide to Applying Human Factors Methods : Human Error and Accident Management in Safety-Critical Systems*, vol. 1. London: Springer London, 2004.

[24] N. Mehta, S. Shinde, N. Tiku, and P. G. Student, “Centralized Database for Android and Web Application,” *Int. J. Innov. Res. Sci. Eng. Technol. (An ISO Certif. Organ.*, vol. 3297, no. 11, 2007.

[25] J. Robert L. Foster, “Database Management Systems For Mobile Devices.” [Online]. Available: people.cs.ksu.edu/~hankley/d764/Slides08/764\_22\_Foster\_DB4Mobile.ppt. [Accessed: 02-Dec-2017].

[26] “Sqlite vs h2 | Databases | Hammer Principle.” [Online]. Available: http://hammerprinciple.com/databases/items/sqlite/h2. [Accessed: 12-Oct-2017].

[27] T. Georgiev and E. Georgieva, “User Interface Design for Mobile Learning Applications.”

[28] Lainey Ying, “#Zomato: Because Life Is Too Short To Search Endlessly/Aimlessly For Good Food - Hype Malaysia.” [Online]. Available: https://hype.my/2015/87821/zomato-because-life-is-too-short-to-search-endlesslyaimlessly-for-good-food/. [Accessed: 05-Oct-2017].

[29] “About | Zomato.” [Online]. Available: https://www.zomato.com/about. [Accessed: 05-Oct-2017].

[30] “Events near you — Last.fm.” [Online]. Available: https://www.last.fm/events. [Accessed: 05-Oct-2017].

[31] “Concerts.” [Online]. Available: https://support.spotify.com/nz/using\_spotify/features/concerts/. [Accessed: 03-Oct-2017].

[32] D. R. Rahadi, “Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android,” *J. Sist. Inf.*, vol. 6, no. 1, pp. 661–671, 2014.

[33] X. Ferré, N. Juristo, H. Windl, and L. Constantine, “usability engineering Usability Basics for,” *IEEE Softw.*, vol. 18, no. 1, pp. 22–29, 2001.

[34] J. Nielsen, “Usability inspection methods,” *Conf. companion Hum. factors Comput. Syst. - CHI ’94*, pp. 413–414, 1994.

[35] C. Banga and J. Weinhold, *Essential Mobile Interaction Design: Perfecting Interface Design in Mobile Apps*. Addison-Wesley Professional, 2014.

[36] “Android alert dialog example.” [Online]. Available: https://www.mkyong.com/android/android-alert-dialog-example/. [Accessed: 12-Oct-2017].

[37] “Event Finder Full Android Application v1.0 » Download Free Premium Scripts, Wordpress Plugins, Mobile Games and Apps.” [Online]. Available: http://www.codelist.cc/mobile/2271-event-finder-full-android-application-v10.html. [Accessed: 29-Nov-2017].

[38] “calltoaction\_logo\_icon-1.png (2025×1533).” [Online]. Available: http://www.calltoactionmedia.com/wp-content/uploads/2015/08/calltoaction\_logo\_icon-1.png. [Accessed: 29-Nov-2017].

[39] “Free vector icons - SVG, PSD, PNG, EPS &amp; Icon Font - Thousands of free icons.” [Online]. Available: https://www.flaticon.com/. [Accessed: 29-Nov-2017].

[40] “What are all the symbols used by computers?” [Online]. Available: https://www.computerhope.com/issues/ch001801.htm. [Accessed: 29-Nov-2017].

[41] “Eventbrite - Discover popular events &amp; nearby fun - Apl Android di Google Play.” [Online]. Available: https://play.google.com/store/apps/details?id=com.eventbrite.attendee. [Accessed: 29-Nov-2017].

[42] “Understanding the difference between WAMP, LAMP, MAMP and XAMPP - CGurukul - Steering Career, Converting Talent.” [Online]. Available: http://www.cgurukul.com/understanding-the-difference-between-wamp-lamp-mamp-and-xampp/. [Accessed: 05-Dec-2017].

[43] “5 of the Best Local WordPress Development Tools Compared.” [Online]. Available: https://www.codeinwp.com/blog/local-wordpress-development-tools/. [Accessed: 05-Dec-2017].