**Nama aplikasi:**

Jurani

**Definisi :**

Aplikasi pendukung permainan jujur atau berani

**Tujuan :**

1. Mendukung permainan jujur atau berani sebagai game ice breaking saat bersama teman-teman
2. Membawa permainan jujur atau berani ke tahap modern dengan menerapkannya ke dalam platform mobile sehingga pemain bisa mendapatkan pengalaman yang terbaru dalam memainkan permainan jujur atau berani.

**Fitur :**

1. Memasukkan jumlah dan nama pemain
2. Melakukan pengundian untuk menentukan siapa yang mendapat tantangan jujur atau berani di antara pemain yang ada
3. Melakukan pengundian terhadap tantangan yang terpilih baik itu tantangan jujur atau berani

**Modul :**

Modul input pemain, modul pengundi pemain, modul pengundi tantangan.

**Software yang digunakan :**

Android Studio

**REFERENCE**

<http://www.ccpe.csulb.edu/spin/media/pdf/jo_spin-20100212_sw_arch_5w1h_.pdf>

<http://www.slideshare.net/dhiangga/modul-4-eselon-4-manajemen-proyek>

**What is jurani?**

(Apa yang mau kita buat, deskripsinya)

Aplikasi yang akan dibuat adalah sebuah aplikasi permainan yang diadaptasi dari sebuah permainan populer di kalangan remaja dan anak muda, yaitu permainan *truth or dare*. Aplikasi berbasis *mobile* ini mempunyai nama Jurani (jujur atau berani). Adapun fitur-fitur dari aplikasi Jurani ini adalah sebagai berikut:

Fitur :

1. Memasukkan jumlah dan nama pemain
2. Melakukan pengundian untuk menentukan siapa yang mendapat tantangan jujur atau berani di antara pemain yang ada
3. Melakukan pengundian terhadap tantangan yang terpilih baik itu tantangan jujur atau berani

**Who is expected to use jurani?**

(Target user dari aplikasi yang kita buat, mungkin juga dibahas siapa aja yang mengembangkan aplikasi ini)

Dalam membuat suatu produk tentunya dibutuhkan target pasar utama yang akan mengonsumsi/ menggunakan produk tersebut, begitu pula dengan pembuatan aplikasi *mobile*. Seperti yang telah disebutkan sebelumnya bahwa permainan ini merupakan permainan yang populer di kalangan remaja dan anak muda, yang artinya kawula muda ialah target utamanya karena tantangan yang diberikan pun lebih cocok ditujukan bagi kisaran umur remaja. Namun, tidak menutup kemungkinan juga digunakan bagi mereka yang masih duduk di bangku sekolah atau bagi mereka yang sudah dewasa, karena permainan ini bersifat hanya untuk bersenang-senang di saat ngumpul bersama.

**Why do we develop jurani?**

Jurani dibuat untuk mendukung kebahagiaan pada pengguna dan kerabatnya dengan memberikan pengalaman bermain *truth or dare* yang modern dengan app Jurani.

Berawal dari kegelisahan ketika momen kumpul bersama keluarga/kolega/shabat/rekan bisnis yang sering kehabisan topik atau para kerabat yang beralih dengan ponselnya, sehingga sering menghasilkan momen yang kurang berkesan. Sehingga kebutuhan akan kegiatan pendukung keakraban *(Ice breaking activity)* muncul, dan disinilah ide Jurani muncul.

ide Jurani adalah aplikasi yang mampu membuat momen berkumpul atau bertemu lebih akrab dengan melakukan permainan bersama. Pengguna dapat menggunakannya bersama sahabat, kolega, keluarga atau rekan bisnis. Sehingga ide tersebut melatar belakangi pembuatan Jurani.

**How do we create jurani?**

(Bagaimana cara mengembangkan aplikasi ini, bagaimana penggunaannya, dll…)

Aplikasi ini dibuat menggunakan metode *prototype* dalam *System Development Life Cycle*-nya. Sesuai dengan metode yang digunakan, maka pembuatan aplikasi dimulai dengan mendefinisikan semua kebutuhan, baik itu kebutuhan *user* maupun kebutuhan *system.*

(proses pengembangan prototype)..

(iterasi)

(feedback dari user untuk menyempurnakan prototype)

dan proses pembuatan dan pengembangan aplikasi menggunakan *software* Android Studio.

**Device Platform**

(Di mana…..di mana kita buat aplikasi ini (?), masih agak bingung)

Aplikasi ini nantinya akan berjalan di platform *mobile* yaitu Android

**When**

(Waktu pengerjaan aplikasi ini)

Pembuatan aplikasi dimulai pada bulan pertama bulan maret 2016, dimulai dengan tahapan melakukan analisis kebutuhan dari aplikasi yang akan dibuat. Pada tahapan analisis kebutuhan ini diperlukan waktu dua minggu untuk menyelesaikannya. Setelah itu dilanjutkan ke tahapan berikutnya yaitu tahap perancangan, tahap ini dibagi menjadi dua yaitu tahap perancangan alur program yang dimulai dari minggu kedua bulan maret sampai dengan minggu kedua bulan april dan tahap perancangan *wireframe* yang juga dimulai dari minggu kedua bulan maret sampai dengan minggu kedua bulan april. Kemudian dilanjutkan ke tahap implementasi yang juga dibagi menjadi dua tahapan, tahap pertama adalah implementasi antarmuka yang dimulai dari minggu ketiga bulan maret sampai dengan minggu ketiga bulan april dan tahap implementasi perangkat lunak dimulai dari minggu ketiga bulan maret sampai dengan minggu ketiga bulan april. Selanjutnya adalah tahap pengujian, tahapan ini berlangsung dua kali, yang pertama pada minggu pertama bulan april dan dilakukan selama seminggu, tahap yang kedua dilaksanakan pada bulan april minggu ketiga. Tahapan terakhir adalah peluncuran aplikasi yang dilakukan pada minggu keempat bulan april.

## **REQUIREMENTS**

Jurani memiliki tujuan untuk memberikan pengalaman bermain truth or dare dengan aplikasi mobile dengan tantangan yang melibatkan pemain lain. Melihat tujuannya, kebutuhan yang dibutuhkan adalah :

1. kemampuan aplikasi untuk menyimpan data pemain
2. kemampuan aplikasi untuk menyimpan data jenis tantangan
3. kemampuan untuk mengacak kedua data tersebut
4. Kemampuan untuk menampilkan tampilan tambahan(*about* dan *help*)

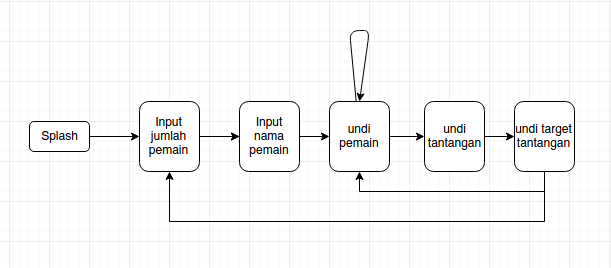
## **ANALYSIS**

Jurani menganalisa kebutuhan game truth or dare khas jurani sehingga constrain dan alur dari fungsi utama yang diperlukan adalah

1. Pengguna diminta untuk memasukkan nama nama pemain
2. Pengguna mengundi nama yang akan ditantang
3. Bila tidak berkenan, pengguna mampu mengundi ulang
4. Pengguna diminta untuk memilih tantangan(Jujur atau Berani)
5. Pengguna mengundi tantangan dari pilihannya
6. Tantangan mengundi target pemain dari tantangan tersebut
7. Tertantang melakukan tantangan(jujur atau berani) terhadap target
8. Kembali ke fase awal

## **DESIGN**

Jurani yang merunut pada analisa, memiliki Struktur navigasi utama sebagai berikut



Secara sederhana rancangan kamus data yang digunakan adalah sebagai berikut

1. Array nama, Array ini digunakan untuk menyimpan nama pemain. Array ini memiliki default length 20, yang mampu dikondisikan lengthnya sesuai dengan jumlah pemain. Array nama ini adalah array yang diisi oleh pengguna.
2. Array tantangan\_jujur, Array ini digunakan untuk menyimpan pertanyaan untuk tantangan jujur. Array tantangan\_jujur memiliki nilai yang absolut yang dalam pengembangannya tidak dapat diubah oleh pengguna
3. Array tantangan\_berani, array ini digunakan untuk menyimpan perintah untuk tantangan berani. Array tantangan\_berani memiliki nilai yang absolut yang dalam pengembangannya tidak dapat diubah oleh pengguna

**PROTOTYPE I dan SPESIFIKASI KEBUTUHAN**

**1. Memasukkan Jumlah Pemain**



**Gambar 1.1.** Tampilan Halaman Memasukkan Jumlah Pemain

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Memasukkan jumlah pemain yang ingin mengikuti permainan. | | | |
| **Policy:** Minimal yang bisa ikut berpartisipasi sebanyak 2 orang dan maksimal sebanyak 20 pemain. | | | |
| **Pre Condition:** Pengguna membuka aplikasi Jurani dan telah melewati halaman *splash*. | | | |
| **Trigger:** Ingin memulai permainan Jujur atau Berani dengan versi yang lebih modern, yaitu menggunakan aplikasi *mobile*. | | | |
| **Major input**  Jumlah Pemain | **Source**  - | **Major outputs**  - | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain | | | |

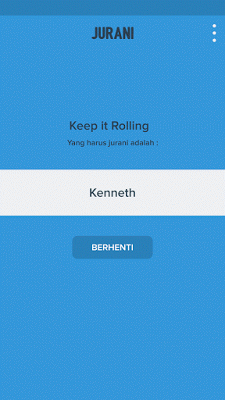
**2. Memasukkan Nama Pemain**



**Gambar 1.2.** Tampilan Halaman Memasukkan Nama Pemain

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Memasukkan nama-nama pemain yang berpartisipasi. | | | |
| **Policy:** Pemain hanya bisa memasukkan nama pemain berupa huruf dan bukan angka. | | | |
| **Pre Condition:** Memasukkan Jumlah Pemain. | | | |
| **Trigger:** Untuk mengetahui nama para pemain yang turut serta mengikuti permainan. | | | |
| **Major input**  Nama Pemain | **Source**  - | **Major outputs**  - | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama-nama pemain sesuai dengan jumlah yang diinput sebelumnya. | | | |

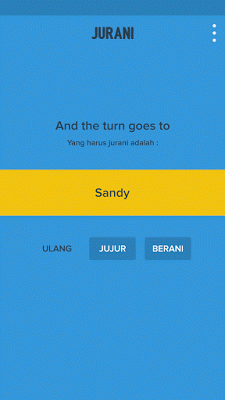
**3. Mengacak Nama Pemain**



**Gambar 1.3.** Tampilan Halaman Pengacakan Nama Pemain

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Melakukan pengacakan pemain untuk menentukan siapa yang mendapat tantangan jujur atau berani di antara *list* pemain yang berpartisipasi. | | | |
| **Policy:** Salah satu pemain harus memberhentikan sistem yang sedang mengacak nama para pemain dengan menekan *button* “Berhenti”. | | | |
| **Pre Condition:** Memasukkan Nama Pemain. | | | |
| **Trigger:** Untuk mengetahui nama yang akan muncul sebagai penerima tantangan. | | | |
| **Major input**  - | **Source**  Nama Pemain | **Major outputs**  Salah Satu  Nama Pemain | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang | | | |

**4. Mengacak Ulang Nama Pemain**



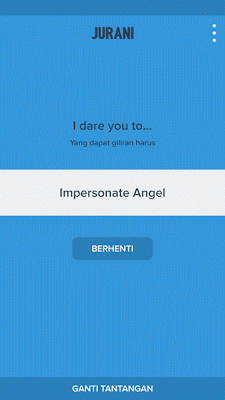
**Gambar 1.4.** Tampilan Halaman Pengacakan Ulang Nama Pemain atau Memilih Tantangan

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Mengacak ulang list nama pemain untuk menentukan nama lainnya yang akan mendapatkan tantangan jujur atau berani. | | | |
| **Policy:** Pemain memilih *button* “Ulang” untuk melakukan pengacakan kembali. Meskipun begitu, masih ada kemungkinan nama yang muncul di sebelumnya akan muncul kembali pada saat memberhentikan pengacakan ulang oleh sistem. | | | |
| **Pre Condition:** Memberhentikan pengacakan nama pemain (muncul nama pemain yang akan ditantang). | | | |
| **Trigger:** Apabila pemain tersebut sudah terlalu sering menerima tantangan. | | | |
| **Major input**  - | **Source**  Nama Pemain | **Major outputs**  Salah Satu  Nama Pemain | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain meminta sistem mengacak ulang nama pemain | | | |

**5. Memilih Tantangan Jujur atau Berani**

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Pemain yang ditantang boleh memilih di antara dua kategori tantangan, yaitu Jujur atau Berani. | | | |
| **Policy:** Pemain hanya boleh memilih Jujur atau Berani dengan menekan salah satu *button* yang tersedia. | | | |
| **Pre Condition:** Memberhentikan pengacakan nama pemain (muncul nama pemain yang akan ditantang). | | | |
| **Trigger:** Apabila pemain tersebut sudah siap untuk mendapatkan tantangan. | | | |
| **Major input**  Pilihan Jujur  atau Berani | **Source**  - | **Major outputs**  Semua List Tantangan Jujur atau Berani | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain memilih tantangan jujur atau berani | | | |

**6. Mengacak Tantangan Jujur atau Berani**



**Gambar 1.5.** Tampilan Halaman Pengacakan Tantangan Berani atau Mengganti Tantangan

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Melakukan pengacakan list tantangan jujur atau berani untuk menentukan tantangan apa yang harus dilakukan (berani) atau dijawab (jujur) serta pengacakan nama pemain yang dijadikan sebagai target untuk menjadi objek yang ditantang. | | | |
| **Policy:**   * Pemain yang ditantang harus memberhentikan sistem yang sedang mengacak list tantangan jujur atau berani dengan menekan *button* “Berhenti”. * Apabila salah satu tantangan sudah terpilih, nama target yang ditantang sudah otomatis teracak oleh sistem. Sehingga, jika ingin mengubah nama targetnya maka tantangan pun harus diacak kembali. | | | |
| **Pre Condition:** Memilih salah satu kategori tantangan, yaitu Jujur atau Berani agar pengacakan list tantangan disesuaikan dengan kategori tersebut. | | | |
| **Trigger:** Untuk mengetahui tantangan yang harus diterima oleh penerima tantangan. | | | |
| **Major input**  - | **Source**   * Semua List Tantangan Jujur atau Berani * Semua Nama Pemain | **Major outputs**   * Salah Satu Tantangan Jujur atau Berani * Nama Target yang Ditantang | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain memilih kategori tantangan, jujur atau berani 7. Sistem mengacak list tantangan sesuai kategori dan nama target sebagai objek tantangan 8. Pemain memberhentikan pengacakan dan muncul satu tantangan serta targetnya | | | |

**7. Mengganti Tantangan Berani ke Tantangan Jujur atau Sebaliknya**

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Pemain yang sebelumnya memilih satu tantangan, misalnya tantangan Berani, dapat berubah pikiran untuk pindah ke tantangan Jujur. | | | |
| **Policy:** Sebelum pengacakan list tantangan, pemain harus terlebih dahulu menekan *button* “Ganti Tantangan” sebelum pemain tersebut menghentikan pengacakan tantangan. Setelah hal tersebut dilakukan, maka akan diarahkan langsung ke halaman pengacakan di kategori tantangan satunya. | | | |
| **Pre Condition:** Sistem melakukan pengacakan list tantangan untuk kategori tantangan tertentu, misalnya tantangan Berani ke tantangan Jujur. | | | |
| **Trigger:** Pemain tiba-tiba berubah pikiran untuk menerima kategori tantangan satunya. | | | |
| **Major input**  - | **Source**  - | **Major outputs**   * Semua List Tantangan Jujur atau Berani * Semua Target yang Ditantang | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain memilih kategori tantangan, jujur atau berani 7. Sistem mengacak list tantangan sesuai kategori dan nama target sebagai objek tantangan 8. Pemain menekan *button* “Ganti Tantangan” 9. Sistem mengacak list tantangan pada kategori lainnya serta nama objek tantangan | | | |

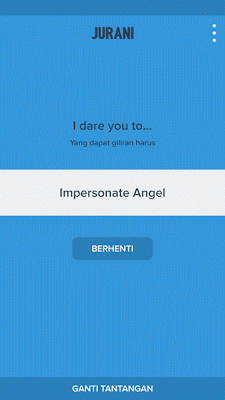
**8. Mengacak Ulang Tantangan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Pemain yang sebelumnya memilih satu tantangan, misalnya tantangan Berani, dapat berubah pikiran untuk pindah ke tantangan Jujur. | | | |
| **Policy:** Sebelum pengacakan list tantangan, pemain harus terlebih dahulu menekan *button* “Ganti Tantangan” sebelum pemain tersebut menghentikan pengacakan tantangan. Setelah hal tersebut dilakukan, maka akan diarahkan langsung ke halaman pengacakan di kategori tantangan satunya. | | | |
| **Pre Condition:** Sistem melakukan pengacakan list tantangan untuk kategori tantangan tertentu, misalnya tantangan Berani ke tantangan Jujur. | | | |
| **Trigger:** Pemain tiba-tiba berubah pikiran untuk menerima kategori tantangan satunya. | | | |
| **Major input**  - | **Source**  - | **Major outputs**   * Semua List Tantangan Jujur atau Berani * Semua Target yang Ditantang | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain memilih kategori tantangan, jujur atau berani 7. Sistem mengacak list tantangan sesuai kategori dan nama target sebagai objek tantangan 8. Pemain menekan *button* “Ganti Tantangan” 9. Sistem mengacak list tantangan pada kategori lainnya serta nama objek tantangan | | | |

**PROTOTYPE II dan SPESIFIKASI KEBUTUHAN**

Pada Prototype kedua ini, secara keseluruhan tidak jauh berbeda dengan Prototype pertama, yang membedakan hanya pada fitur yang terdapat pada halaman Pengacakan Tantangan Jujur atau Berani dan juga penambahan halaman baru, yaitu halaman Pengacakan Nama Pemain yang Menjadi Target Penerima Tantangan

**1. Mengacak Tantangan Jujur atau Berani**



**Gambar 2.1.** Tampilan Halaman Mengacak Tantangan Jujur

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary actor**: *User* yang menggunakan aplikasi Jurani. | | | |
| **Description:** Melakukan pengacakan list tantangan jujur atau berani untuk menentukan tantangan apa yang harus dilakukan (berani) atau dijawab (jujur). | | | |
| **Policy:** Pemain yang ditantang harus memberhentikan sistem yang sedang mengacak list tantangan jujur atau berani dengan menekan *button* “Berhenti”. | | | |
| **Pre Condition:** Memilih salah satu kategori tantangan, yaitu Jujur atau Berani agar pengacakan list tantangan disesuaikan dengan kategori tersebut. | | | |
| **Trigger:** Untuk mengetahui tantangan yang harus diterima oleh penerima tantangan. | | | |
| **Major input**  - | **Source**  Tantangan Jujur  atau Berani | **Major outputs**  Semua List Tantangan Jujur atau Berani | **Destination**  *-* |
| **Major Step Performed**   1. Membuka aplikasi Jurani 2. Memasukkan jumlah pemain 3. Memasukkan nama pemain 4. Sistem mengacak nama pemain 5. Pemain memberhentikan pengacakan dan muncul satu nama pemain yang akan ditantang 6. Pemain memilih kategori tantangan, jujur atau berani 7. Sistem mengacak list tantangan sesuai kategori 8. Pemain memberhentikan pengacakan dan muncul satu tantangan | | | |