

RANCANG BANGUN APLIKASI BARENG BERBASIS ANDROID DAN GOOGLE API

Alvin Nur Luqman, Universitas Ciputra UC Town, Citraland, Surabaya email: aluqman@student.ciputra.ac.id¹

Trianggoro Wiradinata, Universitas Ciputra UC Town, Citraland, Surabaya email: twiradinata@ciputra.ac.id²

ABSTRAK

Pertumbuhan jumlah penduduk mengakibatkan meningkatnya jumlah pengguna layanan transportasi umum maupun alat transportasi pribadi. Sebagai kota *Metroplis*, Surabaya merupakan pusat kegiatan perdagangan barang dan jasa, industri, maupun pemerintahan. Peningkatan *mobilitas* penduduk tersebut menyebabkan peningkatan terhadap penggunaan kendaraan bermotor. Namun, hal tersebut tidak diimbangi dengan penambahan panjang jalan. Pada akhirnya, kondisi tersebut mengakibatkan terjadinya titik kemacetan lalu lintas di sebagian besar jalan di Kota Surabaya termasuk wilayah Citraland dan Universitas Ciputra. Dengan merancang bangun sebuah aplikasi bareng berbasis android dan google API diharapkan dapat menambah sarana transportasi baru bagi masyarakat dengan pemakaian yang mudah yaitu dengan perangkat mobile android yang banyak di gunakan saat ini dan mengurangi titik kemacetan yang disebabkan oleh banyaknya kendaraan pribadi di kota Surabaya.

Melalui aplikasi bareng berbasis android dan google api memungkinkan seseorang untuk membuat barengan, mencari barengan atau bareng. Dengan pencarian lokasi peta dari dan tujuan yang di permudah dengan bantuan google api. Telah dilakukan pengujian pada 10 responden untuk mencoba 7 fitur aplikasi bareng yaitu login, buat barengan, komentar barengan, bareng, deal, panggil dan cari barengan. Hasil yang di dapatkan 10 pengguna tersebut telah berhasil menjalankan fitur tanpa gagal. Maka aplikasi bareng berhasil di rancang bangun dan tidak di dapati kegagalan fitur atau *error*. Rencana pengembangan aplikasi akan diberi sistem keamanan yang memadahi agar pengguna lebih percaya serta memperindah tampilan aplikasi.

Kata Kunci : Bareng, transportasi, Android, aplikasi, Google

1. Pendahuluan

Transportasi merupakan salah satu komponen penting masyarakat dalam melakukan aktivitas sehari-hari. Apalagi transportasi pribadi di era modern ini yang serba canggih dan cepat, banyak masyarakat yang di buru waktu. Oleh karena itu, kemudahan pergerakan dari satu tempat ke tempat yang lain menjadi sangat penting, sehingga dituntut kemampuan untuk mengoptimasi biaya dan waktu.

Pertumbuhan jumlah penduduk mengakibatkan meningkatnya jumlah pengguna layanan transportasi umum maupun alat transportasi pribadi. Sebagai kota *Metroplis*, Surabaya merupakan pusat kegiatan perdagangan barang dan jasa, industri, maupun pemerintahan. Pertumbuhan ekonomi yang pesat menyebabkan tingkat pergerakan penduduk semakin

meningkat. Peningkatan mobilitas penduduk tersebut menyebabkan peningkatan terhadap penggunaan kendaraan bermotor. Namun, hal tersebut tidak diimbangi dengan penambahan panjang jalan. Pada akhirnya, kondisi tersebut mengakibatkan terjadinya titik-titik kemacetan lalu lintas di sebagian besar jalan di Kota Surabaya termasuk wilayah Citraland dan Universitas Ciputra.

Banyak kerugian yang di dapat dari kepadatan lalu lintas, seperti kerugian waktu, uang dan tenaga. Laporan milik Kresnayana Yahya (2011; *Surabaya Macet, Rp. 1 Triliun Menguap*), seorang pengamat dari *ITS Enciety Business Consult*, menyatakan bahwa padatnya lalu-lintas di Surabaya dapat mengakibatkan kerugian sebesar 1 triliun per hari. Hal ini membuktikan pentingnya akses transportasi yang lancar dan lapang di kota Surabaya.

Berbagai macam cara sudah banyak dilakukan oleh berbagai pihak untuk mengurangi kepadatan lalu lintas di kota Surabaya. Seperti penambahan jumlah alat transportasi umum telah banyak dilakukan, namun tidak sepenuhnya efektif masih banyak pengguna layanan transportasi umum harus menunggu lama untuk mendapatkan layanan transportasi. Padahal penyedia layanan transportasi umum seperti taksi sudah menggunakan aplikasi *mobile* untuk menawarkan kemudahan pelanggan dalam pemesanan taksi.

Sebagian besar masyarakat memilih untuk membeli alat transportasi pribadi murah seperti sepeda motor dan mobil murah seperti *LCGC (Low Cost Green Car)*. Hal itu memperparah kepadatan lalu lintas di kota Surabaya. Pemerintah juga tidak tinggal diam untuk mengatasi masalah tersebut, contohnya dengan melakukan pelebaran dan penambahan jalan raya. Namun prosesnya memerlukan waktu yang cukup lama dan sebagian besar menimbulkan titik-titik kepadatan lalu lintas di kota Surabaya.

Berdasarkan latar belakang diatas, maka di rancang aplikasi *mobile BARENG* berbasis Android untuk mengurangi kepadatan lalu lintas.

2. Landasan Teori

2.1 System Development Life Cycle

Software Development Life Cycle yang pertama digunakan secara luas sering disebut sebagai model waterfall. Model waterfall merupakan sebuah model proses ideal dimana setiap tahap selesai sebelum tahap berikutnya dilakukan, dan proyek dikerjakan dari satu tahap ke tahap selanjutnya tanpa iterasi atau tumpang tindih. Model waterfall sering digunakan dalam proyek perangkat lunak dan telah berhasil diterapkan. Terkadang beberapa pengulangan tahap pada model waterfall bisa terjadi, hal ini terjadi disaat terdeteksinya kesalahan.

Software Development Life Cycle (SDLC) sendiri adalah struktur proses yang digunakan dalam pembuatan perangkat lunak. Model waterfall memiliki tahapan yaitu requirement gathering, analysis and design, implementation, testing, dan maintenance. Pembuatan aplikasi pada tugas akhir ini menggunakan berbasis SDLC dengan model waterfall.

Tahap requirement gathering merupakan sebuah proses untuk mengetahui dan memahami kebutuhan dan harapan pengguna. Adanya komunikasi dengan pengguna diperlukan untuk menentukan kebutuhan yang dapat dipenuhi dari perangkat lunak yang akan dibuat. Kebutuhan tersebut harus jelas dan lengkap. Salah satu cara untuk mengetahui kebutuhan dan harapan pengguna adalah dengan melakukan survey. Survey sendiri merupakan teknik umum untuk mengetahui kebutuhan pengguna dari batasan dari sebuah proyek.

Tahap analysis and design adalah proses menentukan komponen, interface, dan data untuk sebuah aplikasi sehingga aplikasi dapat dibuat sesuai keinginan pengguna. Sebelum desain dilakukan analisa akan dilakukan terlebih dahulu menggunakan Unified

Modelling Language. Setelah tahap analisa sudah dilakukan, maka dilanjutkan dengan tahap desain yaitu pembuatan User Interface Design yang merupakan gambaran dari perangkat lunak yang akan dibuat.

Masuk ke tahap implementation, dimana merupakan tahap pengembangan perangkat lunak tersebut. Setelah tahap implementation selesai, maka bisa dilakukan tahap testing untuk mengetahui apakah perangkat lunak dapat bekerja sesuai dengan harapan pengguna, maka tahap deployment dan maintenance yang merupakan kegiatan pengoprasian perangkat lunak dapat dilakukan. Dalam tahap deployment dan maintenance juga akan dilakukan perubahan-perubahan pada aplikasi untuk meningkatkan fungsi aplikasi sesuai harapan pengguna. Namun, tahap deployment dan maintenance tidak akan dibahas pada bab selanjutnya karena batasan tugas akhir ini hanya sampai pada tahap testing. (Fatta, 2007: 24)

2.2. UML

Unified Modelling Language (UML) merupakan sebuah bahasa pemodelan yang menjadi standar dalam membantu pendeskripsian dan perancangan sistem perangkat lunak, terutama perangkat lunak yang dibangun menggunakan object-oriented (OO) (Whitten, 2004). UML dapat digunakan untuk membuat model jenis sistem perangkat lunak, dimana sistem tersebut dapat berjalan pada perangkat keras, sistem informasi dan jaringan manapun serta ditulis dalam bahasa pemrograman apapun. UML terdiri dari notasi-notasi grafis dimana notasi grafis itu sendiri dapat memberikan penjelasan mengenai sistem perangkat lunak tersebut.

Salah satu metodologi desain yang paling umum digunakan dalam OOP saat ini adalah Unified Modelling Language. UML dikembangkan pada awal 1980-an sebagai respon terhadap kebutuhan untuk cara yang standar dan pemodelan sistematis perancangan perangkat lunak object-oriented. Ini terdiri dari serangkaian model tekstual dan grafis dari solusi yang diajukan. Model ini mendefinisikan ruang lingkup sistem, komponen sistem, interaksi pengguna dengan sistem, dan bagaimana komponen sistem berinteraksi satu sama lain untuk menerapkan sistem fungsi. Berikut ini adalah beberapa model umum yang digunakan dalam UML:

1. Use case

Use case diagram adalah teknik untuk mengetahui persyaratan fungsional dari sebuah sistem. Use case diagram bekerja dengan menggambarkan jenis interaksi antara pengguna sistem dan sistem itu sendiri, menyediakan sebuah naratif tentang bagaimana sistem digunakan. Penjelasan/tekstual grafis tentang bagaimana sistem akan berperilaku dari perspektif pengguna. Pengguna bisa saja manusia atau pun sistem lainnya.

2. Sequence diagram

Sebuah model urutan interaksi objek sebagai program dijalankan. Penekanan ditempatkan pada urutan interaksi dan bagaimana mereka berproses dari waktu ke waktu. Diagram interaksi menggambarkan bagaimana sekelompok objek berkolaborasi dalam beberapa sifat.

UML menetapkan beberapa bentuk dari diagram interaksi, dan yang paling umum adalah sequence diagram. Sequence diagram menangkap sifat dari sebuah skenario. Diagram tersebut akan menunjukkan sebuah angka dari contoh objek dan pesan yang telah diberikan antara objek-objek tersebut dalam use case.

3. Activity diagram

Activity diagram merupakan teknik untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja. Dalam banyak cara, activity diagram mempunyai peranan yang hampir sama dengan flowchart, tetapi perbedaan utama activity diagram berbeda dengan notasi flowchart adalah activity diagram mendukung perilaku paralel.

2.3 Android SDK (Software Development Kit)

Android SDK adalah perangkat *API (Application Programming Interface)* yang diperlukan untuk mendesain dan merancang aplikasi pada *platform Android* (Reto Meier, 2010:6) menggunakan bahasa pemrograman *Java*. *Android* adalah subset perangkat lunak untuk ponsel yang meliputi *system operasi, middleware*, dan aplikasi kunci yang dikeluarkan oleh *Google* (Reto Meier, 2010:4). Aplikasi *mobile* *Bareng* dibangun menggunakan bahasa pemrograman *Java*. Secara detail, isi dari *Android SDK* terdiri dari (Reto Meier, 2010:12) :

1. *Android API (Application Programming Interface)* yang merupakan inti dari *SDK Android API* merupakan kumpulan folder untuk mengakses kelas-kelas dan kode milik *Android*. Pustaka ini sama seperti yang digunakan *Google* untuk merancang aplikasi *native* yang sudah ada di dalam *Android*.
2. *Development Tools* yang berguna untuk melakukan *compile* dan *debug* aplikasi.
3. *Android Virtual Device Manager and Emulator* yang merupakan *emulator Android* interaktif dengan berbagai macam *skin*. *Emulator* tersebut berjalan di dalam *AVD (Android Virtual Device)* yang menyimulasikan konfigurasi perangkat keras alat. *Emulator* ini berguna untuk melihat bagaimana tampilan dan kegunaan aplikasi *Android* yang sedang di bangun di dalam peralatan *mobile*.

2.4 Sejarah Sistem Operasi Android

Penemu android adalah Andy Rubin yang lahir pada tanggal 22 Juni 1946 di New Bedford, Amerika Serikat. Andy Rubin bersama dengan Rich Miner, Nick Sears, dan Chris White mendirikan android.inc. Nama "android" berasal dari istilah *Android*, yang mengacu pada sebuah robot yang dirancang untuk melihat dan bertindak seperti manusia. *Android* adalah system operasi mobile yang awalnya dikembangkan oleh androidinc, kemudian android dibeli oleh *Google* pada tahun 2005. *Google* dan *Open Handset Alliance (OHA)* berkolaborasi dalam pengembangan *Android*. Semenjak itu nama penyedia aplikasi *Android* pun

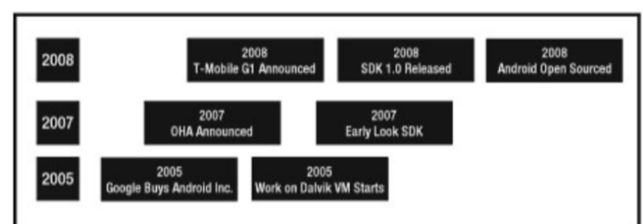
berubah dari *AndroidMarket* menjadi *Google play*. (Yuniar,2011)

Pada tanggal 5 November 2007, kelompok pemimpin industri bersama-sama membentuk *Open Handset Alliance (OHA)* yang diciptakan untuk mengembangkan standar terbuka bagi perangkat *mobile*. *OHA* terdiri dari 34 anggota besar dan beberapa anggota yang terkemuka diantaranya sebagai berikut: *Sprint Nextel®*, *T-Mobile®*, *Motorola®*, *Samsung®*, *Sony Ericsson®*, *Toshiba®*, *Vodafone®*, *Google*, *Intel®* dan *Texas Instruments*.

Android SDK dirilis pertama kali pada 12 November 2007 dan para pengembang memiliki kesempatan untuk memberikan umpan balik dari pengembangan *SDK* tersebut. Pada bulan September 2008, *T-Mobile* memperkenalkan ketersediaan *T-Mobile G1* yang merupakan *smart phone* pertama berbasis *platform Android*. Beberapa hari kemudian, *Google* merilis *Android SDK 1.0*. *Google* membuat *source code* dari *platform Android* menjadi tersedia di bawah lisensi *Apache's open source*.

Google merilis perangkat genggam (disebut *Android Dev Phone 1*) yang dapat menjalankan aplikasi *Android* tanpa terikat oleh berbagai jaringan provider telepon seluler pada akhir 2008. Tujuan dari perangkat ini adalah memungkinkan pengembang untuk melakukan percobaan dengan perangkat sebenarnya yang dapat menjalankan *Android OS* tanpa berbagai kontrak. *Google* juga merilis versi 1.1 dari sistem operasi *Android* pada waktu yang tidak lama. Versi 1.1 dari *Android* tidak mendukung adanya *soft keyboards* dan membutuhkan perangkat yang memiliki *keyboard* secara fisik. *Android* menyelesaikan masalah ini dengan merilis versi 1.5 pada bulan April 2009 dengan sejumlah tambahan fitur seperti kemampuan perekaman media, *widgets*, dan *live folders*.

Versi 1.6 dari *Android OS* dirilis pada bulan September 2009 dan hanya dalam waktu satu bulan versi *Android 2.0* dirilis dan membanjiri seluruh perangkat *Android*. Versi ini memiliki kemampuan *advanced search, text to speech, gestures, dan multi touch*. *Android 2.0* memperkenalkan kemampuan untuk menggunakan *HTML* karena didukung oleh *HTML 5*. Semakin banyak aplikasi berbasis *Android* setiap harinya yang terdapat pada *application store* secara online atau dikenal sebagai *Android Market*.



Gambar 2.1 Gambar Sejarah Sistem Operasi Android

2.5 Google API

Google api adalah sekumpulan aplikasi antarmuka pemrograman (*APIs*) dikembangkan oleh *Google* yang

memungkinkan komunikasi dengan layanan Google dan integrasi mereka ke layanan lain. Contoh pencarian, Gmail, menerjemahkan atau Google Maps (Elian, 2012).

2.6 Java

Menurut Budi Raharjo, Imam Heryanto, Arif haryono (Mudah Belajar Java 2010) java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.7 PHP

Menurut Sibero (2011:49) "PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan". Php disebut juga pemrograman Server Side Programming, hal ini dikarenakan seluruh prosesnya dijalankan pada server. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan open source yaitu pengguna data mengembangkan kode-kode fungsi sesuai kebutuhannya.

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang PHP: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.8 Mysql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Relational Database Management System (RDBMS).

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License) (Arbie, 2004). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih

unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

3. Desain Sitem

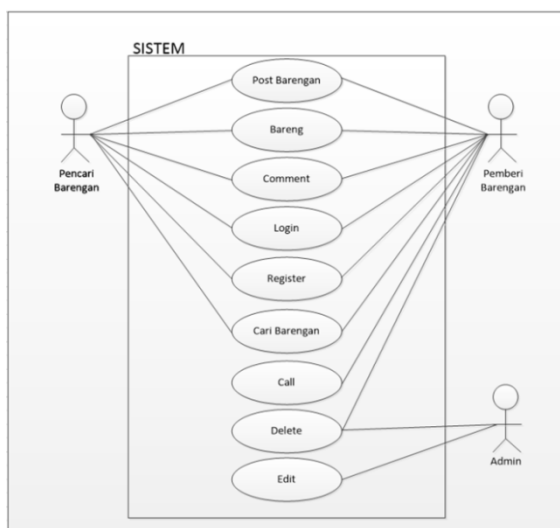
Pada pembahasan analisis dan desain sistem penulis merancang desain aplikasi yang meliputi, *use case diagram*, *activity diagram*, *sequence diagram*, *database design*, *ui mockup*, dan nilai *entrepreneurship*.

3.1 Use Case Diagram

Bareng adalah sebuah aplikasi untuk mengurangi kepadatan lalu lintas dari dan menuju kampus Universitas Ciputra Surabaya. Aplikasi Bareng terbagi menjadi 3 *role*, yaitu pemberi barengan, pencari barengan dan admin seperti dapat dilihat pada gambar 3.1.

Pada aplikasi Bareng pemberi barengan dapat membuat post barengan, bareng, comment, login dengan akun google, register, cari barengan, menghubungi pencari barengan dan menghapus post barengan.

Admin diharuskan untuk *sign in* terlebih dahulu sebelum mengakses halaman admin, admin bertugas untuk mengedit menambah atau menghapus post barengan, data user dan lokasi yang terdapat salah ketik, *error*, kata – kata SARA dan lain sebagainya.



Gambar 3.1. Use Case Diagram.

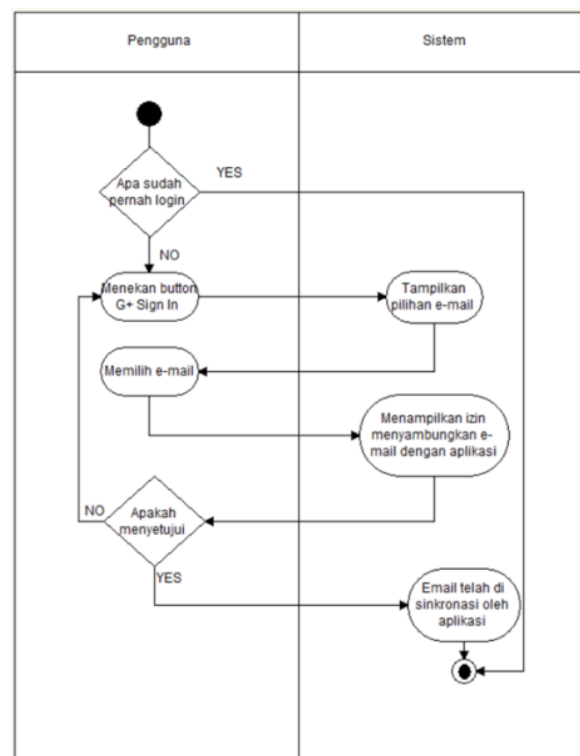
3.2 Activity Diagram

Activity diagram adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung

aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

3.2.1 Activity Diagram Login

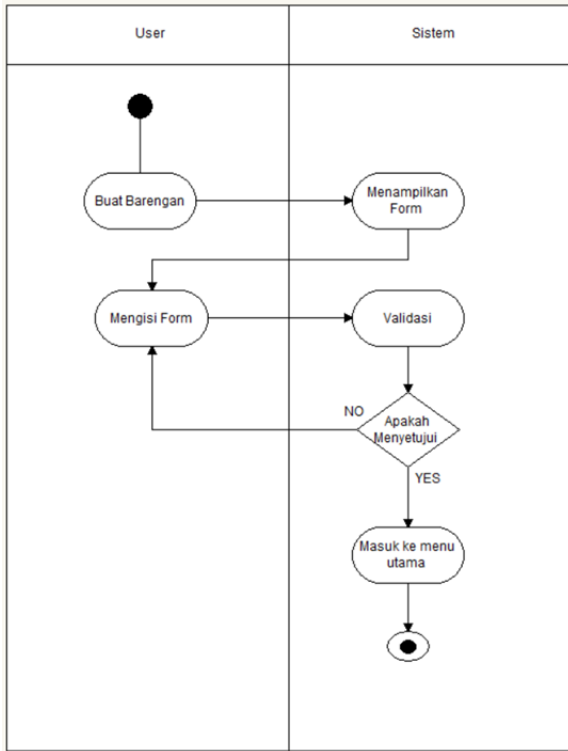
Pada fitur login seperti gambar 3.2, sebelum memasuki aplikasi user diwajibkan untuk login terlebih dahulu. User menekan tombol google sign-in lalu memilih email google yang akan di pakai untuk login. Jika email google sudah di sinkronasi dengan handphone maka pengguna bisa langsung mengakses aplikasi dengan email tersebut.



Gambar 3.2. Activity diagram login

3.2.2 Activity Diagram Buat Barengan

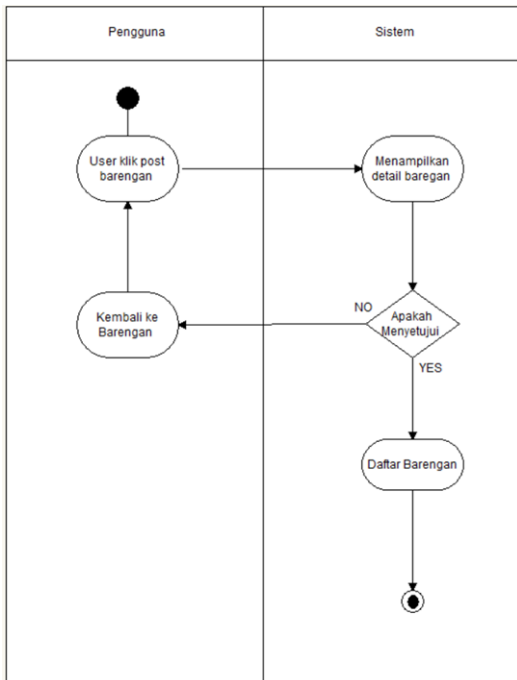
Dalam membuat barengan seperti gambar 3.1., pemberi barengan harus masuk ke menu Buat Barengan, kemudian akan ada *form* yang berisi detail barengan. Setelah melengkapi *form* para pembuat barengan akan menekan *post* yang akan di validasi terlebih dahulu oleh admin, jika ada kolom yang belum di isi maka pembuat barengan harus mengisi kembali form yang di sediakan. setelah di validasi sistem akan menampilkan *post* pada menu utama.



Gambar 3.3. Activity Diagram Buat Barengan.

3.2.3 Activity Diagram Bareng

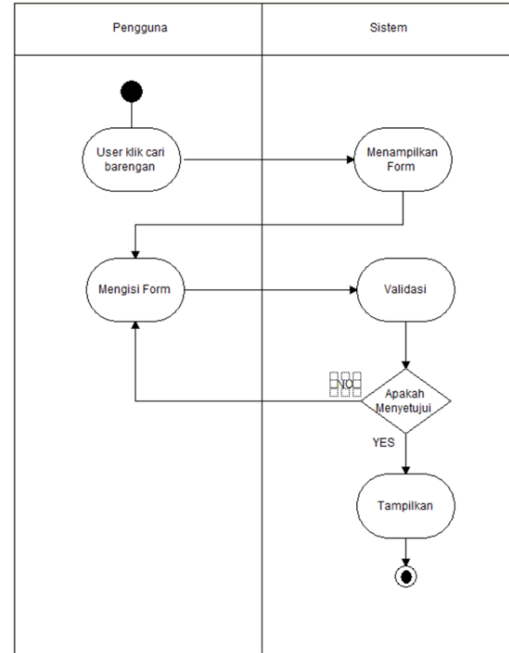
Untuk mencari barengan user masuk ke menu utama lalu memilih *post* barengan. Setelah melihat detail, user akan menekan tombol bareng lalu *post* barengan akan berpindah ke menu daftar barengan.



Gambar 3.4. Activity Diagram Bareng.

3.2.4. Activity Diagram Cari Barengan

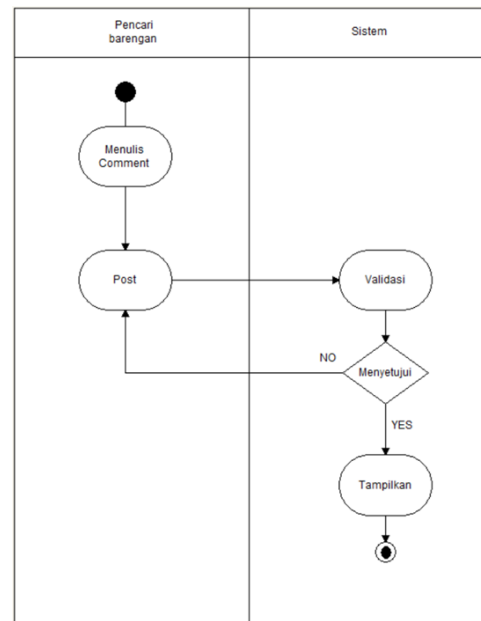
Untuk mencari post Barengan, user masuk menu cari barengan kemudian mengisi form detail. Setelah melengkapi form detail user klik search, maka sistem akan menampilkan hasil post barengan sesuai detail yang dimasukan user.



Gambar 3.5. Activity Diagram Cari Barengan.

3.2.5. Activity Diagram Comment

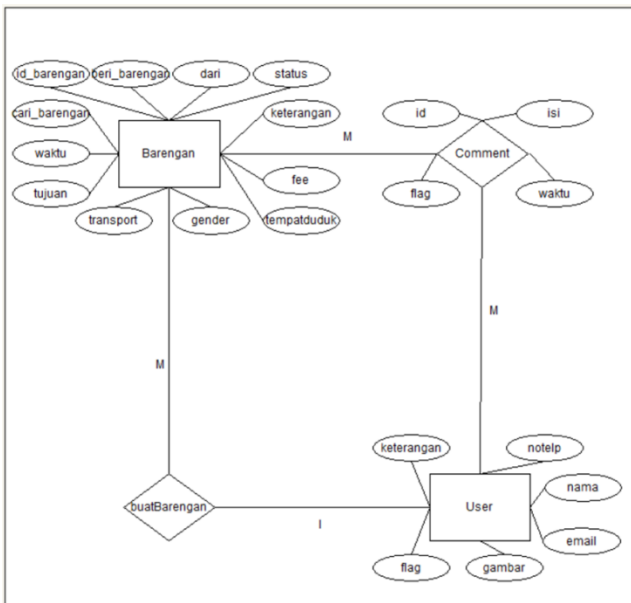
Untuk membuat comment, user menulis komentar pada kolom yang di sediakan. Setelah menulis comment user menekan tombol post, setelah di setuju admin comment akan di tampilkan.



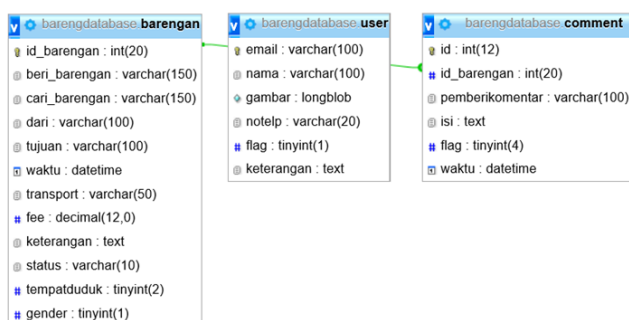
Gambar 3.6. Activity Diagram Comment.

3.2.6 Database Design

Database Design merupakan gambar data-data yang telah di tentukan dan di butuhkan dalam sistem. Untuk memastikan informasi yang di hasilkan terpenuhi dengan baik.



Gambar 3.7. Database Design Aplikasi Bareng.



Gambar 3.8. Database Design Aplikasi Bareng.

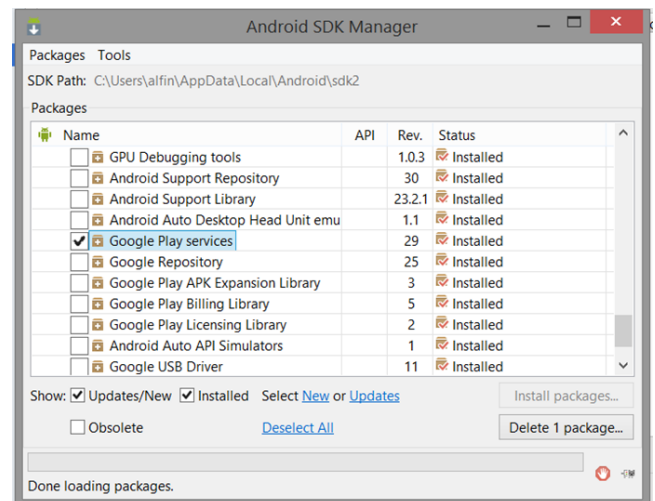
4. Implementasi

4.1 Instalasi Google Play Service

Google Play Service adalah bagian dari Sistem Development Kit (SDK) yang ada dalam Android Studio. Google Play Service merupakan fitur dari Google. Jika Google Play Service sudah terinstal dalam SDK maka hanya perlu melakukan update, dengan cara sebagai berikut :

1. Buka Android Studio.
2. Klik Tools.
3. Klik Android.
4. Klik SDK Manager.
5. Klik Launch Standalone SDK Manager.
6. Centang Google Play Service (Seperti pada Gambar 4.1).
7. Klik Install packages...

8. Accept License.
9. Klik Install.



Gambar 4.1 Android SDK Manager.

4.2 Google Developer Console

Google Developer Console berfungsi untuk mengakses infrastruktur Google, contohnya Gmaps, Google Plus Sign In, Google Calendar dan lain sebagainya. Penulis menggunakan API default dari google untuk pengerjaan tugas akhir ini. Setiap komputer/laptop mempunyai satu kode unik SHA1 fingerprint, dimana kode tersebut bisa didapatkan dengan menggunakan command prompt. Google Developer Console tidak bisa dijalankan tanpa mengetahui kode SHA1 fingerprint, Berikut adalah cara untuk mendapatkan kode SHA1 fingerprint:

1. Bukalah Commad Prompt.
2. Ketik keytool -list -v -keystore "%USERPROFILE%\android\debug.keystore" -alias androiddebugkey -storepass android -keypass android.
3. Mana akan didapatkan kode SHA1 fingerprint, sebagai contoh: 43:91:2E:23:B2:F0:DA:CB:01:9E:FD:B8:9B:G9:A C:C9:26:53:79:18.

Setelah mendapatkan kode SHA1 fingerprint, maka proses selanjutnya adalah membuka situs Google Developer Console, berikut adalah cara menggunakan Google Developer Console.

1. Buka web browser chrome (yang direkomendasikan oleh google).
2. Buka situs <https://console.developers.google.com>.
3. Sign In menggunakan Akun Google.
4. Klik Create Project.
5. Isi nama project.
6. Lengkapi data di tab credential, SHA1 fingerprint dan nama package di laptop/PC anda.
7. Klik Tab Overview.
8. Pilih Google Plus API.

4.3 Implementasi Kode

4.3.1 Implementasi Kode Gradle

Gradle

```
compile
'com.android.support:appcompat-
v7:23.1.1'
compile
'com.android.support:design:23.1.1'
compile 'com.google.android.gms:play-
services:7.5.0'
```

Kode di atas terdapat pada *gradle*, penjelasan kode di atas adalah *gradle* adalah *build automation tool* yang dapat dikonfigurasi melalui DSL berbasis *Groovy*. Penggunaan DSL berbasis *Groovy* menyebabkan *Gradle* lebih fleksibel dan dapat diprogram dengan mudah. Manfaat lain *gradle* adalah untuk perizinan dalam penggunaan library dan pengaturan versi desain. Library berfungsi menampilkan *swiperefresh* dan *circle imageview*. Aplikasi menggunakan versi desain 23.1.1.

4.3.2 Implementasi Kode Manifest

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.INTE
RNET" />
<uses-permission
android:name="android.permission.GET_
ACCOUNTS" />
<uses-permission
android:name="android.permission.USE_
CREDENTIALS" />
<uses-permission
android:name="android.permission.ACCE
SS_NETWORK_STATE" />
<uses-permission
android:name="android.permission.WRIT
E_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.google.android.prov
iders.gsf.permission.READ_GSERVICES"
/>
<uses-permission
android:name="android.permission.ACCE
SS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCE
SS_FINE_LOCATION" />
```

Kode diatas terdapat pada *AndroidManifest.xml*, penjelasan kode di atas sebagai berikut *AndroidManifest* berfungsi untuk memberi akses permission untuk fitur pada aplikasi, menerjemahkan string atau label ke dalam aplikasi yang diperlukan, seperti *string icon launcher*, *app name*, inialisasi dukungan layar perangkat.

Pada manifest aplikasi Bareng *permission* yang digunakan adalah mengakses network, *read* dan *write*

sdcard, *read* dan *write calendar*, *read phone state* untuk keperluan notifikasi.

4.3.3 Implementasi Kode Google Plus Sign In

Sign in adalah proses awal saat menggunakan aplikasi, sign in pada aplikasi ini menggunakan *Google Plus Sign In*. Setelah instalasi *Google Play Service* pada *SDK* dan melakukan konfigurasi pada *Google Developer Console*. Berikut adalah kode dalam fitur sign in.

AndroidManifest.xml

```
useLibrary 'org.apache.http.legacy'
compile 'com.google.android.gms:play-
services:8.3.0'
```

Kode di atas terdapat pada *gradle*, penjelasan kode di atas sebagai berikut *gradle* berfungsi untuk membatasi versi android. Sebagai contoh aplikasi Indonesia Event Portal membatasi android minimal android *jelly bean*. Selain itu *gradle* digunakan untuk menambah perizinan dalam menggunakan library.

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.INTE
RNET" />
<uses-permission
android:name="android.permission.GET_
ACCOUNTS" />

<uses-permission
android:name="android.permission.MANA
GE_ACCOUNTS" />

<uses-permission
android:name="android.permission.USE_
CREDENTIALS" />
<meta-data
android:name="com.google.android.gms.
version"
android:value="@integer/google_play_s
ervices_version" />
```

Kode terdapat pada *AndroidManifest.xml* berfungsi sebagai perizinan agar aplikasi bisa diakses dengan internet, mendapatkan credential Akun Google Plus, dan mendeteksi *gms Google* untuk desain tombol Google Sign In.

Kode terdapat pada *Login.java* penjelasan kode di bawah berfungsi untuk mendapatkan data dari Google yang berupa *string*, data yang didapatkan adalah email, nama, dan alamat foto profil. Kode lengkap terdapat pada lampiran.

Login.java

```

public void onConnected(Bundle arg0)
{
    mShouldResolve = false;
    try {
        if
(Plus.PeopleApi.getCurrentPerson(mGoogleApiClient) != null) {
            Person person =
Plus.PeopleApi.getCurrentPerson(mGoogleApiClient);
            String personName =
person.getDisplayName();
            String email =
Plus.AccountApi.getAccountName(mGoogleApiClient);
            signup(email, personName, "");
            Intent f = new
Intent(Login.this, MenuUtama.class);
            databarengan.email=email;
            startActivity(fux);
            Toast.makeText(getApplicationContext(
),
                "You are
Logged In " + personName,
                Toast.LENGTH_LONG).show();
            } else {

            Toast.makeText(getApplicationContext(
),
                "Couldnt Get
the Person Info",
                Toast.LENGTH_SHORT).show();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        signOutUI();
    }
}

```

5. Pengujian

5.1 User Acceptance Test

User acceptance testing adalah pengujian yang dilakukan oleh pengguna dari sistem untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. *User acceptance testing* pada tugas akhir ini dilakukan kepada 10 pengguna yang berada di Kampus Universitas Ciputra.

5.2 Hasil Pengujian

Setelah melakukan pengujian penulis mencatat keberhasilan yang dilakukan tester dalam melakukan skenario pengujian.

Tabel 1. Hasil Pengujian.

No	Fitur	Total Berhasil	Total Gagal
1	Login	10	-
2	Buat Barengan	10	-
3	Komentar Barengan	10	-
4	Bareng	10	-
5	Deal	10	-
6	Panggil	10	-
7	Cari Barengan	10	-

Berikut adalah komentar saat melakukan pengujian aplikasi :

Tabel 2. Komentar

Pengguna	Komentar
1	Tampilan Kurang menarik
2	Idenya bagus
3	Aplikasi banyak manfaatnya
4	Aplikasi sudah bagus sesuai konsep
5	Fitur untuk memasukkan dari dan tujuan sangat membantu
6	Ide aplikasi bagus
7	Mudah dan simple untuk pemakaiannya
8	Keluarnya aplikasi agak susah
9	Diberi gambar gambar menarik
10	Aplikasi sudah bagus

6. Kesimpulan dan Saran

6.1 Kesimpulan

Dengan mengacu dari perancangan sistem, implementasi, dan pengujian terhadap aplikasi Bareng. Maka didapatkan kesimpulan sebagai berikut:

1. Proses perancangan dan pembangunan aplikasi Bareng telah berhasil dilakukan.
2. Telah dilakukan uji coba kepada 10 responden dengan tingkat keberhasilan yang baik.
3. Seluruh pengujian sudah berjalan sebagaimana mestinya dengan menjalankan fitur-fitur aplikasi Bareng hingga berhasil sebagaimana fungsinya.

6.2 Saran

Berikut adalah beberapa saran saat penulis melakukan pengujian aplikasi untuk pengembangan aplikasi berikutnya:

1. Tampilan aplikasi sebaiknya diperbaiki agar memberikan tampilan yang *user friendly*.
2. Ditambah fitur keamanan yang memudah agar pengguna dapat merasa aman saat memakai aplikasi.

DAFTAR PUSTAKA

- Arbie. (2004). *Manajemen Database Dengan MySQL*. Yogyakarta: ANDI.
- Elian, A., Mazharuddin, A., & Studiawan, H. (2012). *Layanan Informasi Kereta Api Menggunakan GPS, Google Maps dan Android*. Surabaya: ITS.
- Fatta, A., Hanif, & Harjoko, A. (2007). *Merancang bangun sistem presensi karyawan berbasis pengenalan wajah dengan Algoritma Eigenface Studi kasus pada STMIK AMIKOM Yogyakarta*. Yogyakarta.
- Jeffery L. Whitten et al. (2004). *Metode Desain dan Analisis Sistem*. Yogyakarta, IN: ANDI.
- Meier, R., & Meier, R. (2010). *Professional Android 2 application development*. Indianapolis, IN: Wiley.
- Raharjo, B., Heryanto, I., & Haryono, A. (2010). *Tuntunan Pemrograman Java Untuk Handphone dan Alat Telekomunikasi Mobile*. Bandung: Informatika.
- Safaat, N. H. (2012). Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Retrieved from <http://elib.unikom.ac.id/download.php?id=170239>
- Sibero, A. F. (2011). *Kitab Suci Web Programing*. Yogyakarta: MediaKom.
- Supardi, Y. (2011). *Semua Bisa Menjadi Programmer Android Basic*. Jakarta: PT Elex Media Komputindo.
- Yahya, K. (2011). Surabaya Macet, Rp. 1 Triliun Menguap. Retrieved from <http://nasional.news.viva.co.id/news/read/179159-surabaya-macet-rp1-triliun-per-hari-menguap>