

RANCANG BANGUN APLIKASI INFORMASI GEDUNG UNIVERSITAS BRAWIJAYA PADA SMARTPHONE ANDROID

Farid Angga Pribadi, Ismiarta Aknuranda, Wibisono Sukmo Wardhono

Program Studi Teknik Informatika Universitas Brawijaya

anggatpl07@gmail.com

ABSTRACT

Informasi Gedung UB is an application that processes the information that previously has not been used optimally to the information that has value. The main information that is processed in this application is the location of the building in the form of the coordinates. This application shows the location of the existing buildings at UB via mobile devices (Android smartphone). These applications take advantage of the camera function to detect the buildings and use the GPS function to detect the coordinates of the existing buildings at UB. Information displayed using the technology of Augmented Reality (AR). This application is implemented with JAVA programming and IDE Eclipse Helios and was developed by the method of Component Based Software Engineering (CBSE). CBSE method aims to combine existing components and reusable. These applications retrieve the data from the web server that has been tested on the intranet and internet. At the time of the test unit, it was concluded that the unit module of the program meets the functional requirements that have been designed at the design stage. In validation testing can be concluded that the implementation and functionality of the software Informasi Gedung UB has been meeting the needs that have been outlined in the requirements analysis phase. The performance test results show that the performance of the server on the intranet has performed better than the performance of a server on the internet.

Keywords : Information, Augmented Reality, Android, GPS, Coordinat, Webserver

ABSTRAK

Informasi Gedung UB merupakan aplikasi yang mengolah informasi yang sebelumnya belum digunakan secara maksimal menjadi informasi yang memiliki nilai lebih. Informasi utama yang diolah dalam aplikasi ini adalah letak gedung berupa kordinat. Aplikasi ini menunjukkan letak gedung-gedung yang ada di Universitas Brawijaya melalui *mobile device (smartphone Android)*. Aplikasi ini memanfaatkan fungsi kamera untuk mendeteksi gedung-gedung dan menggunakan fungsi gps untuk mendeteksi koordinat dari gedung-gedung yang ada di Universitas Brawijaya. Informasi ditampilkan menggunakan teknologi *Augmented Reality (AR)*. Aplikasi ini diimplementasikan dengan pemrograman JAVA dan IDE Eclipse Helios dan dikembangkan dengan metode *Component Based Software Engineering (CBSE)*. Metode CBSE bertujuan untuk menggabungkan komponen yang sudah ada dan dapat digunakan kembali. Aplikasi ini mengambil data dari *webservice* yang telah diuji pada jaringan intranet maupun internet. Pada saat pengujian unit, disimpulkan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan. Pada pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas perangkat lunak Informasi Gedung UB telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Hasil pengujian performa menunjukkan bahwa pada performa server pada jaringan intranet memiliki performa lebih baik daripada performa server pada jaringan internet.

Kata Kunci : Informasi, Augmented Reality, Android, GPS, Koordinat, Webservice

1. PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi telah diterapkan di berbagai bidang kehidupan, mulai dari bidang kesehatan, pendidikan, pariwisata, bisnis, keamanan dan bidang-bidang lain. Informasi-informasi yang ada diolah dengan teknologi sehingga berguna dalam bidangnya masing-

masing. Teknologi informasi berkembang secara terus menerus. Selalu ada hal baru yang muncul di setiap perkembangannya yang bersifat berkelanjutan.

Perkembangan teknologi informasi saat ini mengarah pada bagaimana menyajikan informasi sehingga pengakses informasi mendapatkan informasi secara mudah, cepat dan akurat dengan

berbagai cara dan berbagai media, salah satunya adalah *Augmented Reality* (AR).

AR adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Benda-benda maya berfungsi menampilkan informasi yang tidak dapat diterima oleh manusia secara langsung. Contohnya, sebuah pola yang mengandung informasi tertentu yang hanya dapat dilihat oleh teknologi AR. Hal ini membuat AR berguna sebagai alat untuk membantu persepsi dan interaksi penggunanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

Universitas Brawijaya (UB) Malang merupakan salah satu universitas negeri di kota Malang. UB memiliki luas area kampus 1.813.664 m² [UBM-12] dan tentunya memiliki banyak gedung. Setiap fakultas maupun jurusan memiliki gedung tersendiri dengan fungsi tersendiri pula. Seperti gedung kuliah, gedung birokrat, gedung kegiatan kemahasiswaan dan lain-lain. Untuk membantu pengunjung UB mengenali gedung-gedung tersebut, dibutuhkan sebuah sistem informasi gedung yang mudah, cepat dan akurat.

Penulis mempunyai gagasan yang dapat memenuhi kebutuhan informasi yang disebutkan sebelumnya. Gagasan ini adalah merancang sebuah aplikasi *mobile* yang dapat mengolah informasi posisi (koordinat) gedung. Dengan menggunakan fungsi kamera pada perangkat *mobile* tersebut, aplikasi ini dapat mendeteksi gedung yang koordinatnya telah disimpan di basis data. Selanjutnya aplikasi ini akan menampilkan informasi yang berkaitan dengan gedung tersebut, misalnya informasi posisi gedung, nama gedung, fungsi utama gedung. Dengan demikian sistem ini akan memberikan informasi kepada pengunjung dengan mudah, cepat dan akurat.

Berdasarkan latar belakang tersebut penulis mengambil judul "Rancang Bangun Aplikasi Informasi Gedung Universitas Brawijaya pada *Smartphone* Android" dengan harapan sistem informasi ini nantinya dapat menjadi sebuah sistem informasi yang dapat memudahkan civitas akademik Universitas Brawijaya pada khususnya dan masyarakat luas pada umumnya.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka dapat dirumuskan permasalahannya yaitu:

1. Bagaimana merancang aplikasi informasi

gedung Universitas Brawijaya pada *smartphone* Android?

2. Bagaimana mengimplementasikan aplikasi informasi gedung Universitas Brawijaya pada *smartphone* Android?
3. Bagaimana membangun sumber data aplikasi informasi gedung Universitas Brwaijaya pada *smartphone* Android?
4. Bagaimana menguji aplikasi informasi gedung Universitas Brawijaya pada *smartphone* Android?

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan lebih terfokus dan tidak terjadi pelebaran topik, maka penelitian tugas akhir ini dibatasi dalam hal:

1. Informasi yang ditampilkan adalah informasi posisi gedung, nama gedung, fungsi utama gedung tersebut.
2. Gedung yang terdeteksi adalah gedung-gedung utama seperti gedung universitas, gedung fakultas, gedung jurusan.
3. Perangkat lunak untuk sistem informasi gedung Universitas Brawijaya menggunakan *AR engine* mixare.
4. Sistem informasi gedung Universitas Brawijaya menggunakan perangkat *mobile* berbasis *Android* minimal versi 2.2 (*froyo*).
5. Sumber data yang digunakan adalah sumber data yang dibangun oleh penulis.

1.4 Tujuan

Tujuan penulisan tugas akhir ini :

1. Merancang aplikasi informasi gedung Universitas Brawijaya pada *smartphone* Android.
2. Mengimplementasikan aplikasi informasi gedung Universitas Brawijaya pada *smartphone* Android.
3. Membangun sumber data aplikasi informasi gedung Universitas Brwaijaya pada *smartphone* Android.
4. Melakukan pengujian aplikasi informasi gedung Universitas Brawijaya pada *smartphone* Android.

1.5 Manfaat

- A. Bagi Pengguna Sistem
 1. Mendapatkan informasi yang cepat, akurat dan *realtime* mengenai gedung-gedung yang ada di Universitas Brawijaya.
- B. Bagi Penulis

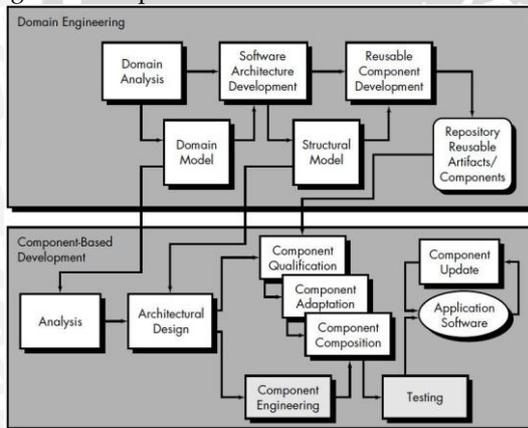
1. Memenuhi syarat kelulusan dari kegiatan perkuliahan di Teknik Informatika Universitas Brawijaya.
2. Mengaplikasikan ilmu yang telah didapat selama perkuliahan.
3. Memperdalam ilmu pengetahuan tentang aplikasi *Augmented Reality*.
4. Memperdalam ilmu pengetahuan tentang sistem operasi Android.

1.6 Tinjauan Pustaka

Pada bab ini berisi pembahasan tentang teori dasar yang berhubungan dengan pengembangan perangkat lunak yang dilakukan.

1.6.1 Component-Based Software Engineering

Component-Based Software Engineering (CBSE) adalah metode pengembangan perangkat lunak yang menekankan pada perancangan dan pembangunan perangkat lunak dengan menggunakan komponen perangkat lunak yang sudah ada dan bersifat dapat digunakan kembali. [PRE-01:721]. CBSE memiliki *software process* yang terdiri dari dua bagian utama yang berjalan paralel, yaitu *domain engineering* dan *component-based development*. Alur *software process* dari CBSE digambarkan pada Gambar 2.1.



Gambar 1. Software process dari CBSE

Sumber : [PRE-01:725]

a. Domain Engineering

Domain Engineering bertujuan untuk mengidentifikasi, membangun, mengumpulkan, dan mengalokasikan sekumpulan komponen perangkat lunak yang dapat dipakai pada pengembangan perangkat lunak pada saat ini dan di masa yang akan datang.

1. Domain Analysis

Pada tahap *domain analysis* akan dilakukan penggalian informasi terkait perangkat lunak yang akan dikembangkan melalui metode analisis kebutuhan sesuai dengan teknik *software analysis*.

Penerapan *domain analysis* pada skripsi ini adalah pada *software requirement specification* (srs).

2. Domain Model

Dalam *Domain Model* akan dilakukan pemodelan terhadap hasil *domain analysis* yang telah dilakukan sebelumnya. Pemodelan dapat dilakukan dengan *unified modelling language* untuk analisis yang bersifat *object-oriented*. *Domain model* diwakili oleh penggunaan UML pada tahap perancangan.

3. Software Architecture Development

Software architecture development merupakan tahap dimana terjadi perancangan pola arsitektur perangkat lunak yang nantinya akan dipakai pada pengembangan perangkat lunak yang akan dibangun. Pada skripsi ini ditunjukkan dalam perancangan arsitektur perangkat lunak.

4. Reusable Components Development

Pada tahap *reusable component development* dilakukan proses identifikasi dan pencarian kebutuhan *software component* yang akan digunakan dalam membangun perangkat lunak. Proses *reusable component development* diwakili oleh *deployment diagram* yang disusun oleh komponen-komponen yang digunakan untuk menyusun aplikasi Informasi Gedung UB.

5. Repository Reusable Artifacts / Components

Repository reusable artifacts / components adalah tahapan dimana komponen – komponen perangkat lunak dikumpulkan dan disimpan dalam bentuk *repository* sehingga siap diimplementasikan pada bagian *component-based development* atau dapat digunakan kembali di masa yang akan datang. Pada skripsi ini diwakili oleh penjelasan tentang komponen-komponen yang menyusun perangkat lunak yang digambarkan pada *deployment diagram*.

b. Component-Based Development

Component-based development bertujuan untuk melakukan pembangunan sistem perangkat lunak dengan menggunakan komponen – komponen perangkat lunak yang bersifat *reusable*.

1. Architectural Design

Tahap *architectural design* merupakan tahapan perancangan arsitektural dari perangkat lunak yang akan dibangun. Perancangan arsitektural ini dilakukan berdasarkan pemodelan kebutuhan fungsionalitas perangkat lunak dan kebutuhan komponen yang telah dilakukan. Dalam perancangan arsitektural ini akan dijelaskan arsitektur perangkat lunak yang akan dibangun, seperti arsitektur penyimpanan data dan arsitektur *class* terkait fungsionalitas perangkat lunak. Umumnya, perancangan arsitektural ini dilakukan dengan *unified modelling language* apabila menggunakan *object-oriented analysis*. Dalam skripsi

ini diwakili oleh *class diagram* dan perancangan basis data.

2. Component Qualification

Pada tahap *component qualification* akan dijelaskan tentang kualifikasi dari komponen perangkat keras dan komponen perangkat lunak yang dibutuhkan dalam pengembangan perangkat lunak yang akan dibangun. Tahap kualifikasi komponen ini bertujuan untuk mengetahui apakah komponen yang digunakan telah sesuai dan memenuhi syarat - syarat dari kebutuhan perangkat lunak yang dibangun. Pada skripsi ini ditunjukkan dalam spesifikasi sistem.

3. Component Adaptation

Dalam tahap *component adaptation* dilakukan proses adaptasi dan modifikasi dari komponen - komponen perangkat lunak yang dipakai sehingga memenuhi kualifikasi kebutuhan komponen dari perangkat lunak yang dibangun. Penggunaan Mixare merupakan penerapan dari *component adaptation*.

4. Component Engineering

Tahap *component engineering* adalah tahap dimana komponen - komponen dari perangkat lunak yang dibangun, dikembangkan secara mandiri. Pengembangan komponen baru ini umumnya dilakukan dengan teknik *object-oriented programming*. Pada tahap ini, diwakili oleh pembuatan *webserver* secara mandiri.

5. Component Composition

Tahap *component composition* adalah tahap utama dimana dilakukan pembangunan perangkat lunak dengan menggunakan komponen- komponen yang sudah ada. Pada tahap ini dilakukan penggabungan komponen - komponen perangkat lunak menjadi satu perangkat lunak yang padu.

6. Testing

Tahap *testing* adalah tahap dilakukannya proses pengujian perangkat lunak yang dibangun menggunakan teknik dan metode pengujian perangkat lunak tertentu. Pengujian dilakukan untuk mengetahui apakah fungsionalitas perangkat lunak sudah benar dan dapat memenuhi kebutuhan pengguna. Tahapan ini diimplementasikan pada bab pengujian.

7. Component Update

Pada tahap ini dilakukan pembaharuan (*update*) komponen perangkat lunak yang telah dibangun untuk meningkatkan kinerja dari perangkat lunak tersebut. Proses tersebut juga termasuk *maintenance* perangkat lunak yang telah didistribusikan. Tahap ini bisa dilihat pada bagian saran namun tidak diimplementasikan pada skripsi ini.

2. METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi rancang bangun aplikasi informasi gedung Universitas Brawijaya pada smartphone Android. Metodologi penelitian yang dilakukan melibatkan beberapa langkah berikut:

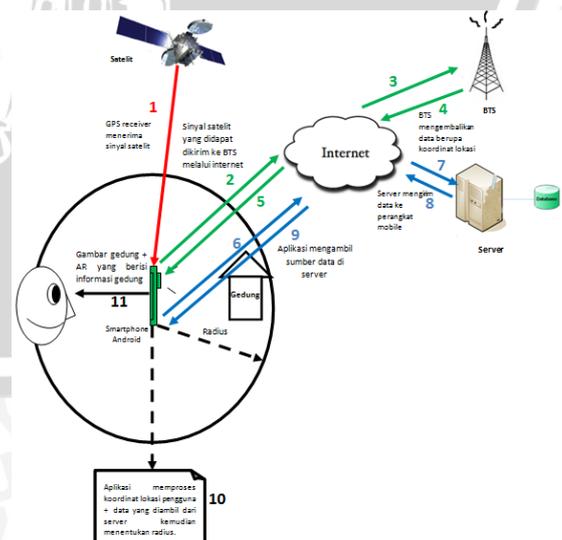
1. Studi Literatur
2. Analisis Kebutuhan
3. Perancangan
4. Implementasi
5. Pengujian dan analisis

Keterkaitan antar langkah-langkah tersebut dapat dilihat pada gambar 2. Setiap langkah ini akan dijelaskan pada sub bab berikutnya setelah deskripsi singkat tentang penggunaan aplikasi informasi gedung Universitas Brawijaya pada smartphone Android.



Gambar 2. Diagram Alir Metodologi Penelitian

2.1 Deskripsi Aplikasi



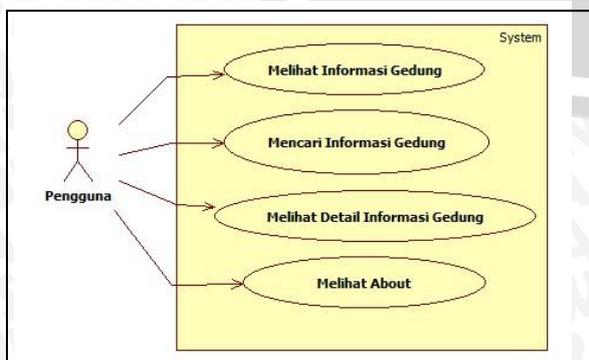
Gambar 3. Prinsip Kerja Aplikasi informasi gedung Universitas Brawijaya pada smartphone Android.

Aplikasi *augmented reality* informasi gedung Universitas Brawijaya merupakan aplikasi yang mengolah informasi yang sebelumnya belum digunakan secara maksimal menjadi informasi yang memiliki nilai lebih. Informasi utama yang diolah dalam aplikasi ini adalah letak gedung berupa kordinat. Aplikasi ini menunjukkan letak gedung-gedung yang ada di Universitas Brawijaya melalui *mobile device* (*smartphone* android). Aplikasi ini memanfaatkan fungsi kamera untuk mendeteksi gedung-gedung dan menggunakan fungsi *gps* untuk mendeteksi koordinat dari gedung-gedung yang ada di Universitas Brawijaya. Informasi ditampilkan menggunakan teknologi *Augmented Reality* (AR).

Aplikasi ini dibuat untuk dapat menghasilkan informasi yang mudah, akurat dan cepat. Kemudahan didapatkan dari *device* yang digunakan yaitu *mobile device* dalam hal ini adalah *smartphone* Android. Banyak orang membawa perangkat ini, dimanapun dan kapanpun. Kecepatan informasi didapatkan dari penggunaan AR sebagai teknologi untuk menampilkan informasi. AR menampilkan informasi secara *realtime*. AR yang digunakan adalah *markerless* AR yang dalam bahasa indonesia diartikan menjadi AR tanpa marker. Hal ini bukan berarti AR tidak memakai *marker* namun menggunakan *marker* yang tak terlihat yang yaitu koordinat lokasi. Pembacaan koordinat lokasi dilakukan saat aplikasi dijalankan dan menghasilkan informasi yang bersifat *realtime*. Keakuratan data yang ditampilkan bergantung pada proses pengumpulan data. Data yang dikumpulkan akan disimpan di *database* yang diletakkan di *server*.

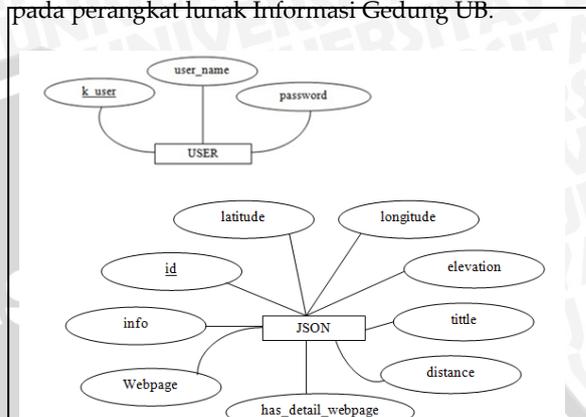
3. PERANCANGAN

Perangkat lunak Informasi Gedung UB memiliki beberapa kebutuhan fungsional yang dimodelkan dalam diagram *use case*. Diagram *use case* perangkat lunak Informasi Gedung UB dibagi menjadi 2, yaitu diagram *use case* aplikasi *client* dan diagram *use case* aplikasi *administrator*. Gambar 4 menggambarkan diagram *use case* aplikasi *client*.



Gambar 4. Diagram *Use Case* Aplikasi *Client*

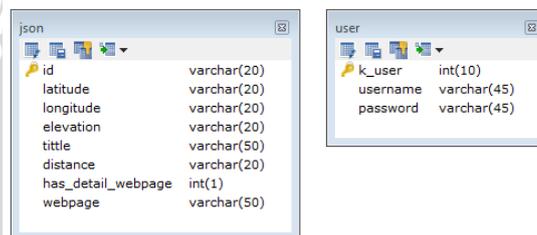
Perangkat lunak Informasi Gedung UB memiliki struktur penyimpanan data yang dimodelkan dalam sebuah diagram *entity relationship*. Gambar 5 menggambarkan diagram *entity relationship* dari struktur penyimpanan data pada perangkat lunak Informasi Gedung UB.



Gambar 5 Diagram *Entity Relationship* Perangkat Lunak Informasi Gedung UB

4. IMPLEMENTASI

Perangkat lunak Informasi Gedung UB diimplementasikan dalam lingkungan sistem operasi Microsoft Windows 7 64-bit dan dalam bahasa pemrograman Java dengan menggunakan IDE Eclipse helios. Tahapan implementasi dilakukan pada perangkat *mobile* Samsung GT5660 dengan *platform* Android. Implementasi penyimpanan data dilakukan dengan *Database Management System* MySQL. Hasil implementasi penyimpanan data ini berupa *script - script* SQL. Gambar 6 menggambarkan diagram ER konseptual dari Aplikasi *Administrator*.

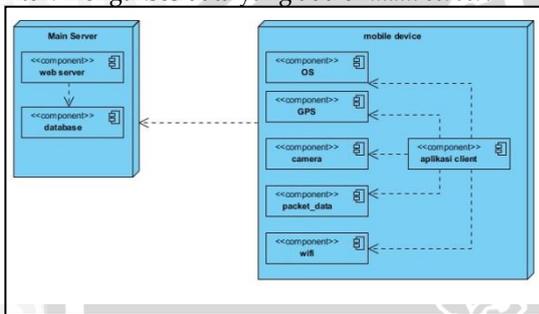


Gambar 6. Diagram ER Konseptual Aplikasi *Administrator*

Gambar 7 menggambarkan komponen-komponen yang menyusun perangkat lunak Informasi gedung UB dan dimana komponen-komponen tersebut di-*deploy*. Perangkat lunak Informasi Gedung UB menggunakan *web hosting* sebagai *main server*. Komponen-komponen yang menyusun *main server* adalah *webserver* dan *database*. *Webserver* digunakan sebagai *web service* untuk



mengirim data yang dibutuhkan. *Web service* yang digunakan adalah JSON. *Database* digunakan sebagai media untuk penyimpanan data. Database yang digunakan adalah *mysql*. *Mobile device* yang digunakan perangkat lunak Informasi Gedung UB adalah Samsung Galaxy Gio dengan OS Android. *Mobile device* terdiri dari beberapa komponen yang dibutuhkan oleh perangkat lunak Informasi Gedung UB, yaitu OS, kamera, GPS, *mobile internet*, dan aplikasi *client*. OS berfungsi sebagai *environment* tempat berjalannya aplikasi *client*. Aplikasi *client* mengakses fungsi kamera untuk menampilkan gedung-gedung yang akan di deteksi. GPS digunakan untuk menentukan posisi pengguna dan mengenali *marker* yang berupa titik koordinat dari gedung. *Mobile internet* digunakan untuk mengakses data yang ada di *main server*.



Gambar 7. Deployment Diagram Informasi Gedung UB.

Fungsi ini untuk melakukan pencarian informasi gedung UB. Setelah masuk menu search, isikan lokasi yang dicari pada *text box* kemudian tekan tombol cari. Setelah itu proses akan berjalan. Gambar 8 merupakan algoritma untuk mencari informasi gedung UB

```

NAMA ALGORITMA Mencari informasi gedung

DEKLARASI
edit : EditText
btnImg : ImageButton
url, key, search : String
i : Intent

DESKRIPSI
Proses :
1. run btnImg.setOnClickListener(new
onClickListener ())
2. url = edit.getText().toString();
3. Intent i = new
Intent(getApplicationContext(),
MixView.class);
4. i.putExtra(key,url).
5. run startActivityForResult(i,0).
6. Class mixview run.
7. If (extras not null) then
a. search = extras.
8. else :
b. search = null.
    
```

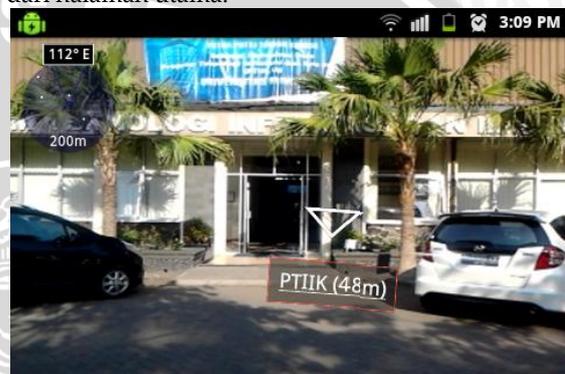
```

9. Request server "http://antonio-
shoes.com/infogedungub/json.php?car
i="+search+"
10. Get JSON data.

Keluaran :
1. View data from server.
    
```

Gambar 8. Implementasi Algoritma Mencari Informasi Gedung

Halaman utama merupakan halaman untuk menampilkan AR. Halaman ini mengakses fungsi kamera yang digunakan sebagai *background layout*. Pada halaman ini terdapat lingkaran radius yang dijadikan ukuran radius koordinat yang di deteksi. Gambar 9 menunjukkan implementasi antarmuka dari halaman utama.



Gambar 9. Implementasi Antarmuka Halaman Utaman

5. PENGUJIAN DAN ANALISIS

Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian validasi, dan pengujian performa. Pada pengujian unit akan digunakan teknik pengujian *White-Box (White-Box Testing)*. Pada pengujian validasi dan performa akan digunakan teknik pengujian *Black-Box (Black-Box Testing)*.

5.1 Pengujian Unit

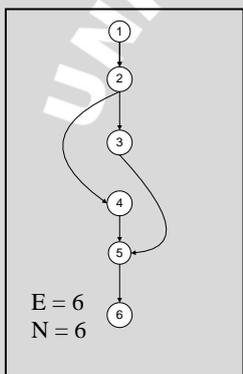
Pada pengujian unit perangkat lunak Informasi Gedung UB digunakan teknik *White-Box Testing* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan.

```

NAMA ALGORITMA Mencari informasi gedung

DEKLARASI
edit : EditText
btnImg : ImageButton
url, key, search : String
i : Intent
DESKRIPSI
Proses :
1. run btnImg.setOnClickListener(new onClickListener ())
2. url = edit.getText().toString();
3. [Intent i = new Intent(getApplicationContext(),
  MixView.class);
4. i.putExtra(key,url).
5. run startActivityForResult(i,0).
6. Class mixview run.
7. If (extras not null) then
  a. search = extras.
8. else :
  a. search = null.
9. Request server "http://antonio-
  shoes.com/infogedungub/json.php?cari="+search+"
10. Get JSON data.
Keluaran :
1. View data from server.
  
```

Gambar 10. Pengujian Unit algoritma mencari informasi gedung.



Gambar 11. Flow Graph Algoritma Mencari Informasi Gedung

Pemodelan ke dalam flow graph yang telah dilakukan terhadap algoritma mencari informasi gedung menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan dua buah basis set dari jalur *independent*, yaitu :

- Jalur 1 : 1 - 2 - 3 - 5 - 6
- Jalur 2 : 1 - 2 - 3 - 4 - 5 - 6

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Variabel search	Data akan ditampilkan	Data akan ditampilkan

	tidak kosong	sesuai dengan variabel search.	sesuai dengan variabel search.
2	Variabel search kosong	Tidak ada yang ditampilkan	Tidak ada yang ditampilkan

Hasil pengujian unit menunjukkan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

5.2 Pengujian Validasi

Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan.

Tabel 1 Hasil Uji Validasi Aplikasi Client

No	Nama Kasus Uji	Status Valid
1.	Menampilkan Informasi Gedung dengan <i>Augmented Reality</i> sebagai Tampilan dalam <i>Mobile Device</i>	Valid
2.	Mencari Informasi Gedung dengan <i>Augmented Reality</i> sebagai Tampilan dalam <i>Mobile Device</i>	Valid
3.	Menampilkan Informasi Detail tentang Gedung yang Dipilih	Valid
4.	Menampilkan Halaman <i>About</i>	Valid

Tabel 2. Hasil Uji Validasi Aplikasi Administrator

No	Nama Kasus Uji	Status Valid
1.	<i>Login</i> Sukses	Valid
2.	<i>Login</i> Gagal	Valid
3.	Menambah Data di <i>Database</i>	Valid
4.	Menghapus Data di <i>Database</i>	Valid
5.	Mengubah Data di <i>Database</i>	Valid

Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas perangkat lunak Informasi Gedung UB telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.3 Pengujian Performa

Pengujian performa dilakukan untuk mengetahui bagaimana performa *server* dari aplikasi dalam mengatasi banyaknya *request* dalam



satuan waktu yang di lakukan secara bersamaan dan dalam jumlah tertentu. Pengujian performa dilakukan menggunakan *tool* yang disebut *apache benchmark*.

Tabel 3. Hasil percobaan *apache benchmark* dengan 100 *request* pada intranet

Request	Concurrency	Request / Second	Transfer Rate
100	5	42,45	137,93 KBps
100	10	78,17	253,99 KBps
100	20	25,14	81,67 KBps
100	40	72,03	234,01 KBps
100	80	57,23	185,96 KBps
Rata-rata		55,004	178,712 KBps

Berdasarkan kasus uji pertama pada Tabel 3 didapatkan hasil sebagai berikut :

- Nilai *transfer rate* minimum sebesar 81,67,04 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 25,14 *request*.
- Nilai *transfer rate* maksimum sebesar 253,99 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 78,17 *request*.
- Dari kelima percobaan yang dilakukan didapatkan nilai *transfer rate* rata – rata adalah sebesar 178,712 KBps dengan dengan *request per second* rata-rata sebesar 55,004 *request*.

Pengujian dilanjutkan dengan mengukur performa *server* yang ada pada *hosting online*, percobaan dilakukan dengan menggunakan 100 *request* dengan jumlah *concurrency* yang berbeda-beda. Tabel 6.37 menunjukkan hasil pengujian *apache benchmark* dengan 100 *request* pada jaringan *Internet*.

Tabel 4. Hasil percobaan *apache benchmark* dengan 100 *request* pada *Internet*

Request	Concurrency	Request / Second	Transfer Rate
100	5	2,06	6,26 KBps
100	10	2,19	6,65 KBps
100	20	2,03	6,15 KBps
100	40	2,05	6,21 KBps
100	80	1,35	4,11 KBps
Rata-rata		1,94	5,89 KBps

Berdasarkan kasus uji pada Tabel 4 didapatkan hasil sebagai berikut :

- Nilai *transfer rate* minimum sebesar 4,11 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 1,35 *request*.
- Nilai *transfer rate* maksimum sebesar 6,65 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 2,19 *request*.
- Dari kelima percobaan yang dilakukan didapatkan nilai *transfer rate* rata – rata adalah

sebesar 5,89 KBps dengan dengan *request per second* rata-rata sebesar 1,94 *request*.

Kasus uji selanjutnya adalah dengan menggunakan jumlah *request* yang lebih banyak yaitu 200 *request* dengan jumlah *concurrency* yang berbeda yaitu mulai dari 5 hingga 80 dengan interval 2n. Tabel 5 menunjukkan hasil pengujian *apache benchmark* dengan 200 *request* pada jaringan intranet.

Tabel 5. Hasil percobaan *apache benchmark* dengan 200 *request* pada intranet

Request	Concurrency	Request / Second	Transfer Rate
200	5	85,47	277,69 KBps
200	10	117,62	382,15 KBps
200	20	62,24	202,2 KBps
200	40	86,62	281,45 KBps
200	80	119,82	389,29 KBps
Rata-rata		94,354	306,556 KBps

Sumber: [Pengujian]

Berdasarkan kasus uji kedua pada Tabel 5 didapatkan hasil sebagai berikut :

- Nilai *transfer rate* minimum sebesar 202,2 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 62,24 *request*.
- Nilai *transfer rate* maksimum sebesar 389,29 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 119,82 *request*.
- Dari kelima percobaan yang dilakukan, didapatkan nilai *transfer rate* rata – rata adalah sebesar 306,556 KBps dengan dengan *request per second* rata-rata sebesar 94,354 *request*.

Pengujian dilanjutkan dengan mengukur performa *server* yang ada pada *hosting online*, percobaan dilakukan dengan menggunakan 200 *request* dengan jumlah *concurrency* yang berbeda-beda. Tabel 6 menunjukkan hasil pengujian *apache benchmark* dengan 200 *request* pada jaringan *Internet*.

Tabel 6 Hasil percobaan *apache benchmark* dengan 200 *request* pada *Internet*

Request	Concurrency	Request / Second	Transfer Rate
200	5	2,26	7,97 KBps
200	10	3,11	9,39 KBps
200	20	3,27	9,93 KBps
200	40	3,46	10,44 KBps
200	80	0	0 KBps
Rata-rata		2,492	7,546 KBps

Berdasarkan kasus uji kedua pada Tabel 6 didapatkan hasil sebagai berikut :

- Nilai *transfer rate* minimum sebesar 0 KBps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 0 *request*.

- b. Nilai *transfer rate* maksimum sebesar 10,44 Kbps dengan jumlah *request per second* yang dapat dilayani *server* sebesar 3,46 *request*.
- c. Dari kelima percobaan yang dilakukan, didapatkan nilai *transfer rate* rata – rata adalah sebesar 7,546 Kbps dengan dengan *request per second* rata-rata sebesar 2,492 *request*.

Dari hasil pengujian dapat ditarik kesimpulan bahwa nilai *transfer rate* dan jumlah *request per second* tidak linier terhadap banyaknya koneksi (*concurrency*) karena tidak memenuhi persamaan linier $y = ax+b$. Ketidaklinieran tersebut disebabkan oleh tidak adanya parameter waktu yang membatasi pada saat pengujian tiap n *concurrency* dan n *request*. Nilai rata-rata dari data hasil pengujian performa dapat disimpulkan bahwa *webservice* pada jaringan *intranet* memiliki performa yang jauh lebih baik daripada performa *webservice* pada jaringan *internet*.

6. KESIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Aplikasi Informasi Gedung UB pada *smartphone* Android dapat digunakan untuk mengenali koordinat gedung – gedung UB yang telah didaftarkan di *datasource* sebelumnya dan menampilkan informasi gedung dengan media *augmented reality* berbasis GPS sesuai dengan perancangan yang telah dibuat.
2. Koneksi data antara *web server* dan aplikasi pada *smartphone* android telah berhasil diimplementasikan dengan metode pengiriman data dari *web server* ke aplikasi *mobile* di android menggunakan metode JSON.
3. Hasil pengujian unit menunjukkan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.
4. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas perangkat lunak Informasi Gedung UB telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.
5. Hasil pengujian nilai *throughput* data pada *server* yang dilakukan menggunakan *apache benchmark* dengan 2 jumlah *request* yang berbeda serta jumlah *concurrency* yang berbeda secara linier menunjukkan bahwa nilai *transfer rate* dan jumlah *request per second* tidak linier terhadap banyaknya koneksi (*concurrency*). Ketidaklinieran tersebut disebabkan oleh tidak adanya parameter waktu yang membatasi pada saat pengujian tiap n *concurrency* dan n *request*.
6. Dari nilai rata-rata data *request per second* dan *transfer rate* yang didapat dari hasil pengujian

dengan parameter *request* dan *concurrency* pada pengujian *webservice* pada jaringan *internet* dan *intranet* dapat disimpulkan bahwa *webservice* pada jaringan *intranet* memiliki performa yang jauh lebih baik daripada performa *webservice* pada jaringan *internet*.

Saran

Saran yang dapat diberikan untuk pengembangan aplikasi Informasi Gedung UB pada *smartphone* android ini antara lain adalah :

1. Berdasarkan permasalahan yang terjadi dalam pengaksesan *datasource* dalam bentuk JSON, sebaiknya digunakan *web server* yang menyediakan hak akses sebagai *root*.
2. Untuk pengembangan lebih lanjut dari aplikasi ini, antarmuka untuk menampilkan informasi dibuat lebih *user friendly*.
3. Aplikasi ini dapat dikembangkan lebih lanjut dengan memberikan hak akses kepada *administrator* untuk menambahkan data gedung (*latitude*, *longitude*, nama gedung, *website*, detail informasi gedung) melalui aplikasi *client*.
4. Aplikasi ini dapat dikembangkan lebih lanjut dengan penambahan *data source* untuk informasi gedung di luar UB seperti kuliner malang, tempat rekreasi dan lain-lain, sehingga dapat digunakan masyarakat luas.

DAFTAR PUSTAKA

- [UBM-12] _____, 2012. Profil UB. http://www.ub.ac.id/id/tentang_ub/profil.html. Tanggal Akses 01 Januari 2012. _____
- [PRE-01] Pressman, Roger S. 2001. *Software Engineering : A Practitioner's Approach, Fifth Edition*. McGraw Hill.