

BAB II

LANDASAN TEORI

2.1 Sistem Informasi Geografis

2.1.1 Konsep Sistem Informasi Geografis

Pada dasarnya, istilah Sistem Informasi Geografis (SIG) merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi dan geografis. Dengan melihat unsur-unsur tersebut, maka jelas SIG merupakan salah satu sistem informasi yang menekankan pada unsur “informasi geografis”. SIG terdiri dari data spasial dan aspasial (Adil, 2017).

Sistem komputer untuk SIG terdiri dari perangkat keras (hardware), perangkat lunak (software) dan prosedur untuk penyusunan input data, pengolahan, analisis, pemodelan (modelling), dan penayangan data geospasial. Sumber data geospasial adalah peta digital, foto udara, citra satelit, tabel statistik dan dokumen lain yang berhubungan. Data geospasial dibedakan menjadi data grafis (atau disebut juga data geometris) dan data atribut (data tematik). Data grafis mempunyai tiga elemen : titik (node), garis (arc) dan luasan (poligon) dalam bentuk vector ataupun raster yang mewakili geometri topologi, ukuran, bentuk, posisi dan arah. Fungsi user adalah untuk memilih informasi yang diperlukan, membuat standar, membuat jadwal pemutakhiran (updating) yang efisien, menganalisa hasil yang dikeluarkan untuk kegunaan yang diinginkan dan merencanakan aplikasi. SIG memudahkan dalam melihat fenomena kebumihantropikan dengan perspektif yang lebih baik.

Secara umum proses SIG terdiri atas tiga bagian (subsistem), yaitu subsistem masukan data, manipulasi dan analisis data, serta menyajikan data. Subsistem masukan data berperan untuk memasukkan data dan mengubah data asli ke bentuk yang dapat diterima dan dipakai dalam SIG. Ada dua macam data dasar geografis, yaitu data spasial dan data atribut. Sedangkan subsistem manipulasi dan analisis data berfungsi menyimpan, menimbun, menarik kembali data dasar dan menganalisa data yang telah tersimpan dalam komputer. Subsistem output data berfungsi menayangkan informasi geografis sebagai hasil analisis data

dalam proses SIG. Informasi tersebut ditayangkan dalam bentuk peta, tabel, bagan, gambar, grafik dan hasil perhitungan.

2.1.2 Manfaat Sistem Informasi Geografis

Fungsi SIG adalah meningkatkan kemampuan menganalisis informasi spasial secara terpadu untuk perencanaan dan pengambilan keputusan. SIG dapat memberikan informasi kepada pengambil keputusan untuk analisis dan penerapan database keruangan (Adil, 2017).

Sistem Informasi Geografis mampu memberikan kemudahan-kemudahan yang diinginkan. Dengan Sistem Informasi Geografis kita akan dimudahkan dalam melihat fenomena kebumihan dengan perspektif yang lebih baik. Sistem Informasi Geografis mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. Sistem Informasi Geografis juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah.

2.1.3 Komponen Sistem (Subsistem) SIG (Sistem Informasi Geografis)

Menurut (Adil, 2017), SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut :

a. Input

Pada tahap *input* (pemasukan data) yang dilakukan adalah mengumpulkan dan mempersiapkan data spasial dan atau atribut dari berbagai sumber data. Data yang digunakan harus dikonversikan menjadi format digital yang sesuai. Proses konversi yang dilakukan dikenal dengan proses digitalisasi.

b. Manipulasi

Manipulasi data merupakan proses *editing* terhadap data yang telah masuk, hal ini dilakukan untuk menyesuaikan tipe dan jenis data agar sesuai dengan sistem yang akan dibuat, seperti: penyamaan skala, pengubahan sistem proyeksi, generalisasi dan sebagainya.

c. Manajemen Data

Tahap ini meliputi seluruh aktifitas yang berhubungan dengan pengolahan data (menyimpan, mengelola, dan menganalisis data) kedalam sistem penyimpanan permanen, seperti: sistem *file server* atau *database server* sesuai kebutuhan sistem.

d. Query

Suatu metode pencarian informasi untuk menjawab pertanyaan yang diajukan oleh pengguna Sistem Informasi Geografis. Pada Sistem Informasi Geografis dengan sistem *file server*, *query* dapat dimanfaatkan dengan bantuan *compiler* atau *interpreter* yang digunakan dalam mengembangkan sistem, sedangkan untuk Sistem Informasi Geografis dengan sistem *database server*, dapat memanfaatkan SQL (*structured query language*) yang terdapat pada DBMS yang digunakan.

e. Analisis

Terdapat dua jenis analisis dalam Sistem Informasi Geografis, yaitu : fungsi analisis spasial, dan analisis atribut. Fungsi analisis spasial adalah operasi yang dilakukan pada data spasial. Sedangkan, fungsi analisis atribut adalah fungsi pengolahan data atribut, yaitu data yang tidak berhubungan dengan ruang.

f. Visualisasi (Data Output)

Penyajian hasil berupa informasi baru atau *database* yang ada baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti dalam bentuk : peta (atribut peta dan atribut data), tabel, grafik dan lain-lain.

2.1.4 Cara Kerja SIG (Sistem Informasi Geografis)

Sistem Informasi Geografis dapat menyajikan *real world* (dunia nyata) pada monitor sebagaimana lembaran peta dapat merepresentasikan dunia nyata diatas kertas. Tetapi, Sistem Informasi Geografis memiliki kekuatan lebih dan fleksibilitas dari pada lembaran pada kertas. Peta merupakan representasi grafis dari dunia nyata, obyek-obyek yang dipresentasikan di atas peta disebut unsur peta atau *map features* (contohnya adalah sungai, taman, kebun, jalan dan lain-

lain). Karena peta mengorganisasikan unsur-unsur berdasarkan lokasi-lokasinya. SIG menyimpan semua informasi deksriptif unsur-unsurnya sebagai atribut-atribut didalam basis data. Kemudian, Sistem Informasi Geografis membentuk dan menyimpannya didalam tabel-tabel (*relasional*) dengan demikian, atribut-atribut ini dapat diakses melalui lokasi-lokasi unsur-unsur peta dan sebaliknya, unsur-unsur peta juga dapat diakses melaluim atribut-atributnya. (Adil, 2017).

2.1.5 Google Map API

Google Map adalah layanan aplikasi dan teknologi peta berbasis web yang disediakan oleh Google secara gartis (bukan untuk kepentingan komersial), termasuk di dalamnya website Google Map (<http://maps.google.com>), Google Ride Finder, Google Transit, dan peta yang dapat disisipkan pada website lain melalui Google Maps API. Saat ini Google Map adalah layanan pemetaan berbasis web yang populer.

Kita dapat menambahkan layanan Google Map ke website kita sendiri menggunakan Google Maps API. Google Maps API dapat ditambahkan ke website kita menggunakan JavaScript. API tersebut menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan, memungkinkan Anda untuk membuat aplikasi peta yang kuat pada website Anda. Pengetahuan yang diperlukan untuk mengembangkan Google Maps API adalah tentang HTML dan JavaScript, sedangkan peta sudah disediakan oleh Google. Jadi kita hanya berkonsentrasi tentang data dan biarkan urusan peta ditangani oleh Google, sehingga dapat menghemat waktu (Adil, 2017).

Tahapan menggunakan Google Map API sebagai berikut :

1. Memasukkan Maps API JavaScript ke dalam HTML.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

2.2 Data Warehouse

Data Warehouse merupakan sebuah gudang data yang berisi data dalam jumlah besar dan digunakan untuk proses analisa dan pembuatan laporan yang dibutuhkan perusahaan (Dahlan, 2017).

Data warehouse mempunyai empat buah karakteristik yaitu : (Dahlan, 2017)

1. Subject Oriented

Data warehouse diorganisasikan ke dalam banyak subject yang utama seperti customer, product, dan sales. Data warehouse tidak berkonsentrasi pada kegiatan operasional sehari-hari dan proses transaksi pada suatu organisasi, data warehouse fokus terhadap modeling dan analisis data untuk pengambilan keputusan, oleh karena itu, data warehouse menghasilkan pandangan sederhana dan ringkasan terhadap subject tertentu dan mengabaikan data yang tidak relevant terhadap proses pendukung keputusan.

2. Integrated

Data Warehouse biasanya dibangun dengan mengintegrasikan sumber data yang berbeda-beda, seperti relational database, flat files , dan online transactional. Teknik data cleaning dan data integration digunakan untuk memastikan konsistensi dalam konvesi penamaan, struktur pengkodean, ukuran atribut, dan sebagainya.

3. Non-volatile

Data warehouse secara fisik memisahkan pengumpulan data dari aplikasi data yang ditemukan dalam operational environment. Di dalam pemisahan data warehouse tidak memerlukan proses transaksi, recovery, dan concurency control mechanism. Biasanya hanya membutuhkan dua operasi dalam mengakses data, initial loading data dan access of data.

4. Time variant Data disimpan untuk memberikan informasi dari perspektif history (misalnya, 5-10 tahun terakhir). Setiap struktur kunci dalam data warehouse berisi baik secara implisit maupun eksplisit elemen waktu.

2.3 Unified Modelling Language (UML)

UML yang merupakan singkatan dari Unified Modelling Language adalah sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek. (Rosa & Shalahudin, 2018).

UML dapat juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berbasis objek. UML pertama kali dikembangkan pada pertengahan tahun 1990an dengan kerjasama antara James Rumbaugh, Grady Booch dan Ivar Jacobson, yang masing-masing telah mengembangkan notasi mereka sendiri di awal tahun 1990an. (Rosa & Shalahudin, 2018).

2.3.1 Komponen-komponen UML

UML mendefinisikan diagram-diagram berikut ini: (Rosa & Shalahudin, 2018).

a. Use case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

Tabel 2.1 Simbol *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri

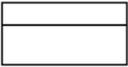
			(<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).

10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
----	---	-------------	---

b. Class Diagram

- 1) Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).
- 2) Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Tabel 2.2. Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

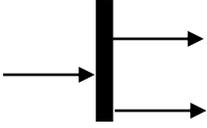
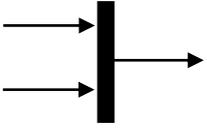
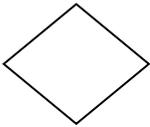
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

c. Activity Diagram

- 1) Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.
- 2) Activity diagram merupakan state diagram khusus, yang sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu, activity diagram tidak menggambarkan perilaku internal sebuah sistem dan interaksi antar subsistem, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Tabel 2.3 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1	<i>Start State</i> 	<i>Start state</i> adalah sebuah kondisi awal sebuah <i>object</i> sebelum ada perubahan keadaan. Start state digambarkan dengan sebuah lingkaran solid.

2.	<i>End State</i> 	<i>End state</i> adalah menggambarkan ketika objek berhenti memberi respon terhadap sebuah event. <i>End state</i> digambarkan dengan lingkaran solid di dalam sebuah lingkaran kosong.
3.	<i>State/Activities</i> 	<i>State</i> atau <i>activities</i> menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya.
4.	<i>Fork (Percabangan)</i> 	<i>Fork</i> atau percabangan merupakan pemisalah beberapa aliran konkuren dari suatu aliran tunggal.
5.	<i>Join (Penggabungan)</i> 	<i>Join</i> atau penggabungan merupakan penggabungan beberapa aliran konkuren dalam aliran tunggal.
6.	<i>Decision</i> 	<i>Decision</i> merupakan suatu logika aliran konkuren yang mempunyai dua cabang aliran konkuren.

d. Sequence Diagram

Sequence diagram secara grafis menggambarkan bagaimana objek berinteraksi antara satu sama lain melalui pesan pada sebuah use case atau operasi.

Tabel 2.4 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.4 Basis Data

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. (Fathansyah, 2018).

Istilah-istilah yang digunakan dalam basis data:

- 1) *File* : merupakan kumpulan dari atribut *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda dalam data *value*-nya.

- 2) *Record* : merupakan kumpulan dari elemen-elemen yang saling berhubungan atau berkaitan menginformasikan tentang *entry* secara lengkap.
- 3) *Field* : merupakan sekumpulan tanda-tanda yang berbentuk kesatuan tersendiri, merupakan bagian terkecil dari *record* dan bentuknya unik dijadikan *field* kunci yang dapat mewakili *record*-nya.
- 4) *Entity* : merupakan tempat kejadian atau konsep yang informasikan direkam.

2.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *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. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau 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 (Fathansyah, 2018).

2.6 PHP (Hypertext Preprocessor)

PHP diciptakan oleh Rasmus Lerdorf, seorang pemrogram C yang handal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada homepagenya. Rasmus adalah seorang pendukung open source. Karena itulah ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan PHP 1.0

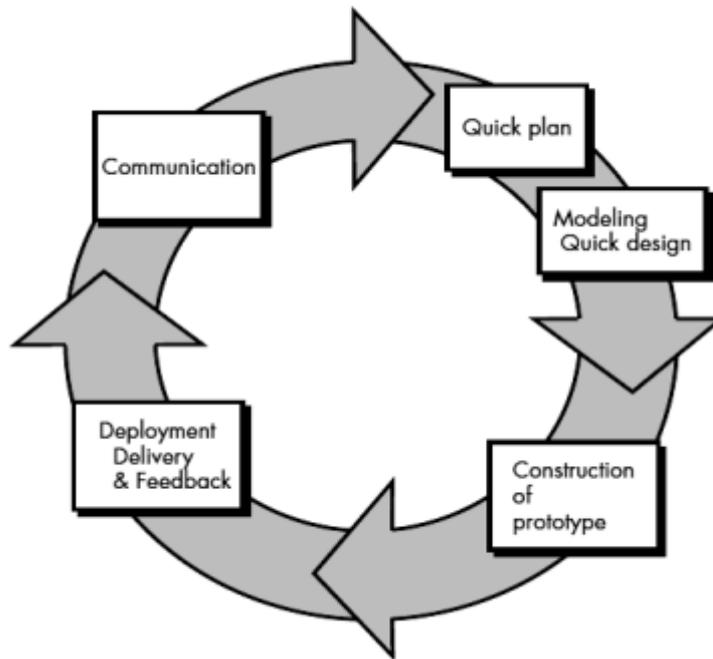
dan menerbitkan PHP 2.0. PHP mudah dibuat dan cepat dijalankan, PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows 98, Windows XP, Windows NT, dan Macintosh (Rerung, 2018).

Kelebihan PHP

1. PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

2.7 Metode Pengembangan Perangkat Lunak Menggunakan Metode *Prototype*

Metode penelitian yang digunakan penulis didasarkan pada metoda pengembangan perangkat lunak prototype yang memiliki tiga tahap. Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa & Shalahudin, 2018). Alasan penulis mengambil metode pengembangan sistem prototype karena pengguna dapat dengan mudah beradaptasi dengan sistem yang dikembangkan sesuai dengan kebutuhan pemakai. Selain itu pada saat tahap evaluasi sistem, jika terdapat revisi pada sistem yang dirancang, programmer tidak harus mengulang tahapan dari awal.



Gambar 2.1 Prototyping Model

(Sumber : Rosa & Shalahudin, 2018)

Berikut penjelasan tahapan metoda prototype yang digunakan dalam penelitian ini.

1. *Communication.*

Penulis menganalisis sistem dengan mewawancarai pihak-pihak fasilitas dan pengguna sebagai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem yang dibuat.

2. *Quick Plan*

Analisis sistem bekerja sama dengan spesialis informasi lain pihak, menggunakan satu atau lebih peralatan prototyping untuk mengembangkan sebuah prototype. (misalnya dengan menambahkan tampilan yang diinginkan pemakai dalam sistem baru seperti menu, interface, fitur, dan database).

3. *Modelling Quick Design*

Analisis membuat perancangan sistem untuk mengembangkan prototipe dengan menggunakan UML, sebagai dasar perancangan aplikasi. Dalam hal

ini programmer menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai database.

4. *Construction Of Prototype*

Tahapan ini adalah tahapan yang dilakukan setelah kegiatan analisi dan perancangan. Pada bagian ini akan dijelaskan kegiatan-kegiatan yang dilakukan pada tahap pengkodean (coding) sistem operasional, implementasi pembuatan program (programming) dan pengujian (testing).

5. *Deployment delivery & feedback*

Penulis yang menguji sistem baru tersebut dan melakukan uji coba terhadap beberapa calon pengguna sehingga penulis dapat menentukan apakah sistem baru dapat diterima. Pada tahap ini pemakai memberi masukan kepada analis apakah sistem dapat diterima. Jika ya sistem baru yang telah diuji dan terima oleh pengguna aplikasi, jika tidak, langkah 4 dan 5 diulangi.