

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitiannya, Rahmi dkk[4] mengembangkan Sistem Informasi Bimbingan Belajar Pada Go Smart Banjarbaru Berbasis *Website*. Sistem ini digunakan khusus menangani masalah administrasi dan pendaftaran. Selain itu terdapat jadwal yang dapat digunakan oleh siswa. Sistem ini dibangun berbasis *website* dengan menggunakan bahasa pemrograman PHP dan MySQL.

Penelitian yang dilakukan oleh Jannah dkk[5] mengenai Rancang Bangun Sistem Informasi Bimbingan Belajar Berbasis *Website* (Studi Kasus : Lembaga Bimbingan Belajar Tadica). Sistem ini digunakan untuk mempermudah siswa dalam mengakses *download* soal latihan, melakukan ujian secara online, dan melakukan percakapan di fitur *chat room*. Sistem ini dibangun berbasis *website* dengan menggunakan bahasa pemrograman PHP dan MySQL.

Penelitian sebelumnya juga dilakukan oleh Hermawan dkk[6] dengan judul "Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis *Website* (Studi Kasus : Yayasan Ganesha Operation Semarang)" tahun 2016. Sistem ini digunakan untuk mempermudah proses penyampaian informasi jadwal kegiatan belajar mengajar kepada pengajar dan siswa melalui *web*. Sistem ini dibangun dengan menggunakan metode *System Development Life Cycle* (SDLC).

Penelitian lain terkait dengan sistem informasi bimbingan belajar adalah "Rancang Bangun Sistem Informasi Akademik Bimbingan Belajar (SIBIJAR) yang dilakukan oleh Hidayat dkk[7] dari Universitas Bina Sarana Informatika , STMIK Nusa Mandiri pada tahun 2019. Sistem ini digunakan untuk proses pengolahan data dan pengarsipan serta penyampaian informasi bimbingan belajar. Sistem ini dibangun dengan menggunakan metode *Waterfall*.

Setelah mengevaluasi penelitian sebelumnya, penulis akan membuat sistem yang berjudul "Sistem Informasi bimbingan Minat Baca dan belajar Anak AIUEO berbasis *Website* (Studi Kasus : biMBA AIUEO Unit Maos Kidul)". Sistem ini terdapat perbedaan dengan penelitian yang dilakukan sebelumnya. Sistem yang akan dikembangkan menyediakan fasilitas pengelolaan data motivator dan murid, hasil perkembangan murid, media penyampaian materi oleh motivator, fitur menambahkan video oleh orang

tua, dan fitur penyampaian pesan dari orang tua ataupun sebaliknya. Sistem yang akan dikembangkan menggunakan bahasa pemrograman PHP dan MySQL berbasis *website* menggunakan metode *waterfall*.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan^[4]. Sistem informasi terdiri dari beberapa komponen diantaranya sebagai berikut :

1. Komponen *input*

Input mewakili data yang masuk kedalam sistem informasi. *Input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan "*tool box*" dalam sistem informasi, Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari *hardware* untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya[8].

2.2.2 Bimbingan Minat Baca dan belajar Anak (biMBA)

AIUEO

biMBA AIUEO adalah lembaga pendidikan anak usia dini informal yang didirikan pada tahun 1998 dengan dukungan dari Yayasan Perkembangan Anak Indonesia (YPAI). Yayasan menciptakan metode yang dianggap cocok untuk melatih dan mengembangkan minat baca anak.

Disisi lain, arti kata biMBA adalah bimbingan Minat Baca dan Belajar Anak, sedangkan AIUEO dipilih karena kelima huruf ini merupakan huruf vokal. Huruf “AIUEO” adalah huruf sederhana yang digunakan dimana-mana, bahkan bayi bisa menyuarakan “AIUEO”. Sesuai dengan namanya, biMBA AIUEO bukan les atau kursus membaca yang berorientasi pada hasil atau kemampuan, tetapi lebih mendorong minat membaca dan belajar anak-anak sejak usia dini.

biMBA ini memiliki metode pembelajaran tersendiri dalam proses pembelajarannya, untuk mengembangkan minat baca kepada peserta didiknya biMBA AIUEO menggunakan metode *Fun Learning* dan *Small Step System*. *Fun Learning* yaitu menciptakan suasana kelas yang menyenangkan agar anak dapat belajar dan bermain di sekolah sesuai keinginannya, bukan belajar dengan paksa. *Small Step System* berarti bahwa seorang anak belajar sesuai kemampuannya mulai dari yang sederhana yaitu mempelajari huruf vokal sampai ke tahap yang berikutnya[9].

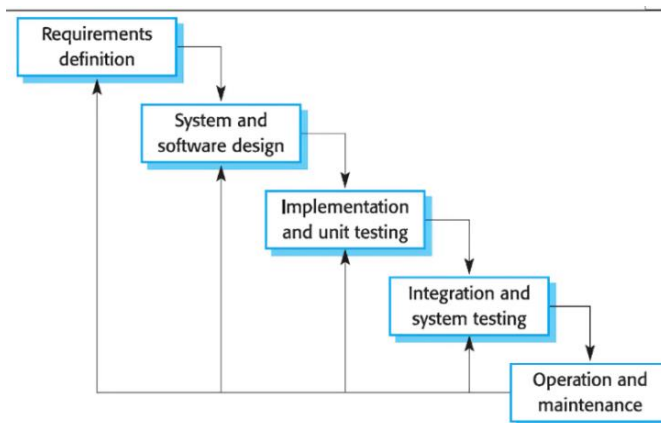
2.2.3 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak berasal dari 2 kata yaitu *Software* (Perangkat Lunak) dan *Engineering* (Rekayasa). Perangkat lunak (*software*) adalah *source code* pada suatu program atau sistem. Perangkat lunak tidak hanya dokumentasi terhadap *source code* tapi juga dokumentasi terhadap sesuatu yang dibutuhkan selama pengembangan, instalasi, penggunaan dan pemeliharaan sebuah sistem. *Engineering* atau rekayasa adalah aplikasi terhadap pendekatan sistematis yang berdasar atas ilmu pengetahuan dan matematis serta aplikasi tentang produksi terhadap struktur, mesin, produk, proses atau sistem[10].

Rekayasa Perangkat Lunak sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa Rekayasa Perangkat Lunak tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan “semua aspek produksi” pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari Rekayasa Perangkat Lunak[11].

A. Metode Pengembangan Sistem

Metode yang digunakan untuk pengembangan sistem yang akan dibuat yaitu metode *waterfall*. Metode *waterfall* adalah “model klasik yang bersifat sistematis, berurutan dalam membangun *software*”. Model ini merupakan model satu arah yang dimulai dari tahapan persiapan sampai perawatan. Tahapan ini meliputi perencanaan, mendesain sistem, implementasi, verifikasi dan perawatan.



Gambar 2.1 Pemodelan *Waterfall*

Metode *Waterfall* memiliki tahapan seperti berikut :

A. *Requirement Analysis and definition*

Tahapan pengembangan sistem diperlukan komunikasi yang bertujuan untuk memahami kebutuhan sistem yang diharapkan pengguna. Dalam tahap ini, juga dianalisis untuk mendapatkan data yang diperlukan.

B. *System and software design*

Spesifikasi kebutuhan dari tahap sebelumnya diterjemahkan ke sebuah perancangan perangkat lunak maupun perangkat keras dengan membentuk arsitektur sistem secara keseluruhan.

C. *Implementation and unit testing*

Tahap *implementation and unit testing* merupakan tahap pemrograman. Pada tahap ini, juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan.

D. *Integration and system testing*

Setelah seluruh unit dikembangkan dan diuji di tahap sebelumnya maka selanjutnya diintegrasikan dalam sistem secara keseluruhan. Kemudian selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

E. *Operation and Maintenance*

Pada tahap ini, pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

B. Metode Pengujian Sistem

Metode pengujian sistem yang digunakan adalah metode *black box testing*. Metode *black box testing* merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas atau kegunaan dari perangkat lunak tersebut. Tujuan dari pengujian *black box testing* ini yaitu untuk mengetahui atau menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi.

Kategori kesalahan yang diuji oleh *black box testing* :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Ciri-ciri *black box testing* :

1. Black box testing berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software* dan *output* yang dihasilkan aplikasi.

Black box testing melakukan pengujian tanpa pengetahuan detail struktur *internal* dari sistem atau komponen yang dites.

2.2.4 Pemograman Berorientasi Objek

Metodologi berorientasi objek adalah strategi pembangunan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis[12].

Metodologi berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan tersruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktifitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai diulang kembali untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

2. Kecepatan pengembangan

Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan

Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola mungkin sering berubah-ubah.

4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak




Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.



Pada pemrograman berorientasi objek, UML digunakan untuk pemodelan sistem. UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifacts* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

A. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2.1 Simbol dan Fungsi *Use Case Diagram*


No	Simbol	Fungsi
1.	 <i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi <i>actor</i> ini bisa berupa orang, perangkat keras atau mungkin juga obyek lain dalam sistem yang sama.
2.	 <i>Include</i>	Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
3.	 Association	Menggambarkan navigasi antar <i>class</i> , berupa banyak obyek lain yang berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.




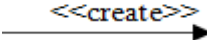
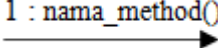
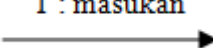
4.	 <i>System Boundary</i>	<i>System Boundary</i> yaitu batasan sebuah sistem.
5.	 <i>Use Case</i>	<i>Use case</i> menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, <i>use case</i> hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.

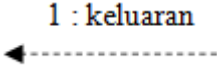
B. *Sequence Diagram*

Sequence diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Sequence diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

Tabel 2.2 Simbol dan Fungsi *Sequence Diagram*

No	Simbol	Fungsi
1.	<i>Actor</i> 	<i>Actor</i> yaitu Orang, poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.





2.	<p><i>Lifeline</i></p> 	<i>Lifeline</i> yaitu menyatakan kehidupan suatu objek.
3.	<p><i>Object</i></p> 	<i>Object</i> yaitu Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Waktu aktif yaitu Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.	<p>Pesan tipe <i>create</i></p> 	Suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe <i>call</i></p> 	Suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri.
7.	<p>Pesan tipe <i>send</i></p> 	Suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	<p>Pesan tipe <i>return</i></p>	Suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan



	suatu kembalian ke objek tertentu.
---	------------------------------------

C. *Class Diagram*

Class Diagram atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi, dan relasinya. Diagram kelas digambarkan dengan bentuk kotak. Simbol dan keterangan *class diagram* seperti pada tabel 2.3.

Tabel 2.3 Simbol dan Fungsi *Class Diagram*

No	Simbol	Fungsi
1.	 <i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
2.	 <i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.	 <i>Class</i>	Kelas pada struktur sistem.
4.	 <i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

5.	 <i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
6.	 <i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.


2.2.5 Basis Data



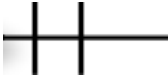
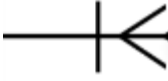

Basis Data atau *Database* adalah himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi data yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi.

A. ERD (*Entity Relationship Diagram*)

Pengertian ERD adalah suatu komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau *Entity Relationship Diagram* menggambarkan data dan hubungan antar data secara global dengan menggunakan *Entity Relationship Diagram*. Simbol-simbol yang digunakan dalam ERD dijelaskan pada Tabel 2.4 Simbol ERD (*Entity Relationship Diagram*) sebagai berikut :

Tabel 2.4 Simbol ERD (*Entity Relationship Diagram*)

No	Keterangan
1. Relasi/Hubungan 	Hubungan yang terjadi antara 1 entitas atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan untuk mengetahui jenis hubungan yang ada antara 2 file.

<p>2. Atribut</p> 	<p>Atribut ialah karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas. Atribut memiliki bentuk lingkaran lebih tepatnya elips.</p>
<p>3. Alur</p> 	<p>Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Dan berbentuk garis.</p>
<p>4. <i>One to One</i></p> 	<p><i>One to One</i>, digunakan untuk menghubungkan antar entitas dengan hubungan satu ke satu</p>
<p>5. <i>One to Many</i></p> 	<p><i>One to Many</i> atau <i>Many to One</i>, digunakan untuk menghubungkan antar entitas dengan hubungan satu ke banyak atau sebaliknya</p>
<p>6. Entitas (<i>Entity</i>)</p> 	<p>Entitas ialah suatu objek yang dapat dibedakan dengan objek lainnya. Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.</p>

Relasi antar table sebagai berikut:

1. Hubungan *One-to-One*, masing-masing tabel hanya terdapat satu data yang saling berhubungan.
2. Hubungan *One-to-Many*, berelasi dengan banyak *record* pada tabel yang lain.
3. Hubungan *Many-to-Many*, banyak *record* pada sebuah tabel berhubungan dengan banyak *record* pada tabel yang lain.

2.2.6 Flowchart



Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya programmer menerjemahkannya ke bentuk program dengan bahasa pemrogram.


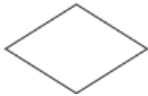


Simbol yang dipakai dalam *flowchart* dibagi menjadi 3 kelompok, yaitu :

1. *Flow direction symbols*
Digunakan untuk menghubungkan simbol satu dengan yang lain (*connecting line*).
2. *Processing symbols*
Menunjukkan jenis operasi pengolahan dalam suatu proses atau prosedur.
3. *Input / Output symbols*
Menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang dipakai antara lain :

Tabel 2.5 Simbol dan Fungsi Flowchart

No	Simbol	Nama dan Fungsi
1.		<i>Flow Direction Symbol</i> yaitu simbol yang digunakan untuk menghubungkan antar simbol
2.		<i>Process Symbol</i> yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.

3.		<i>Manual Operation Symbol</i> yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
4.		<i>Decision Symbol</i> yaitu simbol yang menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya tau tidak.
5.		<i>Terminal Symbol</i> yaitu symbol yang menyatakan permulaan atau akhir suatu program.
6.		<i>Input atau Output Symbol</i> yaitu simbol yang menyatukan proses input atau output tanpa tergantung jenis peralatannya.

2.2.7 Kuesioner

Kuesioner merupakan alat pengumpulan data primer dengan metode survei untuk memperoleh opini responden. Kuesioner dapat didistribusikan kepada responden dengan cara : (1) langsung oleh peneliti (mandiri); (2) dikirim lewat pos (*mailquestionair*);(3) Dikirim lewat komputer misalnya surat elektronik (*e-mail*). Kuesioner dikirimkan langsung oleh peneliti apabila responden relative dekat dan penyebarannya tidak terlalu luas. Kuesioner dapat digunakan untuk memperoleh informasi pribadi responden sikap, opini, harapan dan keinginan responden. Idealnya semua responden mau mengisi atau lebih tepatnya memiliki motivasi untuk menyelesaikan pertanyaan ataupun pernyataan yang ada pada kuesioner penelitian. Apabila tingkat respon (*respon rate*) diharapkan 100% artinya semua kuesioner yang dibagikan kepada responden akan diterima kembali oleh peneliti dalam kondisi yang baik dan kemudian akan dianalisis lebih lanjut.