BAB II

TINJAUAN PUSTAKA

2.1. Profil Instansi

1. Nama : Pondok Pesantren Manahijul Huda
2. Badan Hukum : Heri Hendriyana,SH.,MH No.219 tanggal 23 Nopember 2013
3. No Statistik Pontren : 5.0.0.3.32.06.0515
4. Alamat : Kp. Tejamaya Ds. Sukaraja Kec. Rajapolah Kab Tasikmalaya Jawa Barat Kode Pos 46155
5. Pendiri : KH. Falahudin Ahmad ML
   KH. Agus Maman
6. Lembaga Pendidikan : Takhosus, SMP-SMA Terpadu

2.1.1. Sejarah SMP-SMA Manahijul Huda

1. Nama Pendiri

Manahijul Huda didirikan Oleh KH.Falahudin Ahmad Malihudin besetrta istri Hj.Ai Romlah bersama sama dengan ayahanda KH.Agus Maman Al Muhyidin.

Berawal dari *himmatun aliyah* para pendiri pondok pesantren Manahijul Huda diawali pembangunan diatas tanah beberapa bata atau tumbak, berkat pertolongan Alloh kian hari kian berkembang sampi saat ini .

2. Tahun Berdiri

Berawal dari Pernikahan Falahudin AM (KH.Falahudin AM) Dengan Ai Romlah(Hj.Ai Romlah) pada Hari Senin tanggal 1 Mei 2000 dan satu minggu setelah menikah kedatangan santri sebanyak 11 orang, waktu itu tempat pemondokannya masih di madrasah & rumah (ditengah kawasan padat penduduk)

Dikarenakan santri dari hari kehari yang kian bartambah maka pada tahun 2001 mulai peletakan batu pertama di tengah pesawahan dan mulai dihuni pada

tahun 2002 sekalipun pada saat itu kondisi bangunan masih dalam kondisi 50% dan pondok itu diberi nama Manahijul Huda.

2.1.2. Visi dan Misi

Berikut adalah visi dan misi SMP-SMA Terpadu Manahijul Huda [5]:

2.1.2.1. Visi

Pesantren MANAHIJUL HUDA sebagai pusat lembaga islam yang menyiapkan pemimpin-pemimpin masa depan.

2.1.2.2. Misi

1. Tertarbiyahnya seluruh santri MANAHIJUL HUDA dengan pembinaan islam yang benar (mu'min Muttaqin).
2. Tertanamnya nilai-nilai islami dalam jiwa santri. sehinnga mampu mengaplikasikan dalam kehidupan sehari-hari.(Muslim Muhsinin) .
3. Menggali dan mengembangkan potensi yang dimiliki santri untuk peningkatan kualitas diri dan kehidupan yang mandiri.

2.1.3. Logo SMP-SMA Manahijul Huda



Gambar 2. 1 Logo Pondok Pesantren Manahijul Huda

2.1.4. Struktur Organisasi SMP-SMA Manahijul Huda

Pelindung	: - MUSPIKA KEC. RAJAPOLAH
	- KEP. DESA SUKARAJA
	- PUNDUH KMP. TEJAMAYA
Penasehat	: - KH.NONOP HANAFI
	- KH. E. AHMAD HIDAYAT
	- KH. ABDUL AJIZ
Sesepuh Pontren	: - KH.AGUS MAMAN AL-MUHYIDIN
Pimpinan Umum	: - KH.FALAHUDIN AHMAD ML
Sekretaris Pusat	: - IPANG FIRMANSYAH
Bendahara Pusat	: - HJ. AI ROMLAH
Seksi-seksi Kesantrian	
Ro'is 'Am	: - UST.UAN MASRUHAN, S.Pd.I
Wakil Rois 'Am	:- UST. YAYAN HARDIANSYAH
Sekretaris	:- UST.AHMAD SETIAWAN
Bendahara	: - UST.ANDI HIDAYAT
Kabag Pendidikan	: - UST. USEP AHMAD YUSUF,S.Pd.I
Kabag Humas	: - UST. ADE FITRIANA MUTTAQIN
Kabag Keamanan	: - UST. JAJANG AHMAD
Kabag Kebersihan	: - UST. GUS TENI
Tata Usaha	:- UST.TATANG RONI ARYA

2.2. Landasan Teori

Landasan teori merupakan kumpulan dari beberapa definisi dan konsep yang disusun secara sistematis mengenai variabel-variabel dalam penelitian. Landasan teori ini akan menjadi acuan dasar penulis dalam melakukan penelitian melalui uraian-uraian secara teoritis.

2.2.1. Sistem Pembayaran

Sistem pembayaran adalah sistem yang mencakup seperangkat aturan, lembaga, dan mekanisme yang dipakai untuk melaksanakan pemindahan dana guna memenuhi suatu kewajiban yang timbul dari suatu kegiatan ekonomi. Lantas, apa saja komponen dari sistem pembayaran? sudah barang tentu harus ada alat pembayaran, ada mekanisme kliring hingga penyelesaian akhir (*settlement*). Selain itu juga ada komponen lain seperti lembaga yang terlibat dalam menyelenggarakan sistem pembayaran. Termasuk dalam hal ini adalah bank, lembaga keuangan selain bank, lembaga bukan bank penyelenggara transfer dana, perusahaan *switching* bahkan hingga bank sentral [6]

2.2.1.1. Evolusi Alat Pembayaran

Alat pembayaran boleh dibilang berkembang sangat pesat dan maju. Kalau kita menengok kebelakang yakni awal mula alat pembayaran itu dikenal, sistem *barter* antar barang yang diperjual-belikan adalah kelaziman di era pra moderen. Dalam perkembangannya, mulai dikenal satuan tertentu yang memiliki nilai pembayaran yang lebih dikenal dengan uang. Hingga saat ini uang masih menjadi salah satu alat pembayaran utama yang berlaku di masyarakat. Selanjutnya alat pembayaran terus berkembang dari alat pembayaran tunai (*cash based*) ke alat pembayaran nontunai (*non cash*) seperti alat pembayaran berbasis kertas (*paper based*), misalnya, cek dan bilyet giro. Selain itu dikenal juga alat pembayaran *paperless* seperti transfer dana elektronik dan alat pembayaran memakai kartu (*card-based*) (ATM, Kartu Kredit, Kartu Debit dan Kartu Prabayar) [6].

2.2.1.2. Alat Pembayaran Tunai

Alat pembayaran tunai lebih banyak memakai uang kartal (uang kertas dan logam). Uang kartal masih memainkan peran penting khususnya untuk transaksi

bernilai kecil. Dalam masyarakat moderen seperti sekarang ini, pemakaian alat pembayaran tunai seperti uang kartal memang cenderung lebih kecil dibanding uang giral. Pada tahun 2005, perbandingan uang kartal terhadap jumlah uang beredar sebesar 43,3 persen.

Namun patut diketahui bahwa pemakaian uang kartal memiliki kendala dalam hal efisiensi. Hal itu bisa terjadi karena biaya pengadaan dan pengelolaan (*cash handling*) terbilang mahal. Hal itu belum lagi memperhitungkan inefisiensi dalam waktu pembayaran. Misalnya, ketika Anda menunggu melakukan pembayaran di loket pembayaran yang relatif memakan waktu cukup lama karena antrian yang panjang. Sementara itu, bila melakukan transaksi dalam jumlah besar juga mengundang risiko seperti pencurian, perampokan dan pemalsuan uang.

Menyadari ketidak-nyamanan dan inefisien memakai uang kartal, BI berinisiatif dan akan terus mendorong untuk membangun masyarakat yang terbiasa memakai alat pembayaran nontunai atau *Less Cash Society* (LCS) [6].

2.2.1.3. Alat Pembayaran Nontunai

Alat pembayaran nontunai sudah berkembang dan semakin lazim dipakai masyarakat. Kenyataan ini memperlihatkan kepada kita bahwa jasa pembayaran nontunai yang dilakukan bank maupun lembaga selain bank (LSB), baik dalam proses pengiriman dana, penyelenggara kliring maupun sistem penyelesaian akhir (*settlement*) sudah tersedia dan dapat berlangsung di Indonesia. Transaksi pembayaran nontunai dengan nilai besar diselenggarakan Bank Indonesia melalui sistem BI-RTGS (*Real Time Gross Settlement*) dan Sistem Kliring. Sebagai informasi, sistem BI-RTGS adalah muara seluruh penyelesaian transaksi keuangan di Indonesia.

Bisa dibayangkan, hampir 95 persen transaksi keuangan nasional bernilai besar dan bersifat mendesak (*urgent*) seperti transaksi di Pasar Uang AntarBank (PUAB), transaksi di bursa saham, transaksi pemerintah, transaksi valuta asing (*valas*) serta *settlement* hasil kliring dilakukan melalui sistem BI-RTGS. Pada tahun 2010, BI-RTGS melakukan transaksi sedikitnya Rp174,3 triliun per hari.

Sedangkan transaksi nontunai dengan alat pembayaran menggunakan kartu (APMK) dan uang elektronik masing-masing nilai transaksinya hanya Rp8,8 triliun per hari yang dilakukan bank atau LSB.

Melihat pentingnya peran BI-RTGS dalam sistem pembayaran nasional, sudah barang tentu harus dijaga kontinuitas dan stabilitasnya. Bila sesaat saja sistem BI-RTGS ini ngadat atau mengalami gangguan jelas akan sangat menganggu kelancaran dan stabilitas sistem keuangan di dalam negeri. Hal itu belum memperhitungkan dampak material dan nonmaterial dari macetnya sistem BI-RTGS tadi. Untuk itulah BI sangat peduli menjaga stabilitas BI-RTGS yang dikategorikan sebagai *Systemically Important Payment System* (SIPS). SIPS adalah sistem yang memproses transaksi pembayaran bernilai besar dan bersifat mendesak (*urgent*). Adalah wajar saja apabila Bank Indonesia sangat peduli menjaga kestabilan SIPS dengan mengelola risiko, desain, kehandalan teknologi, jaringan pendukung dan aturan main dalam SIPS. Selain SIPS dikenal pula *System Wide Important Payment System* (SWIPS), yaitu sistem yang digunakan oleh masyarakat luas. Sistem Kliring dan APMK termasuk dalam kategori SWIPS ini. BI juga peduli dengan SWIPS karena sifat sistem yang digunakan secara luas oleh masyarakat. Apabila terjadi gangguan maka kepentingan masyarakat untuk melakukan pembayaran akan terganggu pula, termasuk kepercayaan terhadap sistem dan alat-alat pembayaran yang diproses dalam sistem.

Perlu diketahui bahwa BI bukan semata peduli akan terciptanya efisiensi dalam sistem pembayaran, tapi juga kesetaraan akses hingga ke urusan perlindungan konsumen. Yang dimaksud terciptanya sistem pembayaran, itu artinya memberi kemudahan bagi pengguna untuk memilih metode pembayaran yang dapat diakses ke seluruh wilayah dengan biaya serendah mungkin. Sementara yang dimaksud dengan kesetaraan akses, BI akan memperhatikan penerapan asas kesetaraan dalam penyelenggaraan sistem pembayaran. Sedangkan aspek perlindungan konsumen dimaksudkan penyelenggara wajib mengadopsi asas-asas perlindungan konsumen secara wajar dalam penyelenggaraan sistemnya [6].

2.2.2. *Flowmap*

Flowmap atau bagan alir adalah bagan yang menunjukan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukan dengan jelas.
3. Harus ditunjukan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Adapun simbol-simbol yang sering digunakan dalam flowmap dapat dilihat pada daftar simbol flowmap.

Flowmap digunakan dalam pembangunan aplikasi sistem pembayaran menggunakan *Radio Frequency Identification* (RFID) di *boarding school* pada SMP-SMA Terpadu Manahijul Huda untuk menganalisis presedur manual atau yang sedang berjalan untuk diterapkan kedalam sistem.

2.2.3. Metodologi Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Pada saat ini, metode berorientais objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentranformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility* dan sebagainya) dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan Produktivitas

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

2. Kecepatan Pengembangan

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

3. Kemudahan Pemeliharaan

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

4. Adanya Konsistensi

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

5. Meningkatkan Kualitas Perangkat Lunak

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

2.2.3.1. Analisis Berorientasi Objek

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi objek, apakah kebutuhan yang ada dapat diimplementasikan menjadi sistem berorientasi objek.

Metode OOA saat ini yang paling banyak diginakan adalah dengan menggunakan *tools* UML (*Unifield Modeling Language*) yang akann dijelaskan pada sub bab UML.

2.2.3.2. Perancangan Berorientasi Objek

Perancangan berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke perancangan permodelan agar lebih mudah diimplementasikan dengan pemrograman berorientasi objek.

Permodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat permodelan berorientasi objek, diantaranya adalah UML. Pada tahap ini biasanya kendala dan permasalahan pembangunan sistem dapat dikenali.

OOA dan OOD dalam proses yang berulang-ulang sering memiliki batasan yang samar sehingga kedua tahapan ini sering juga disebut OOAD (*Object Oriented Analysis dan Design*).

2.2.3.3. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek atau *Object Oriented Programming* (OOP) adalah sebuah cara membangun perangkat lunak dengan menggunakan beberapa objek yang akan saling berinteraksi satu sama lain dengan menggunakan bahasa pemrograman berorientasi objek seperti Java, C#, C++, Smalltalk, Eiffel dan PHP.

2.2.3.4. Konsep Dasar Berorientasi Objek

Berikut adalah konsep dasar pemrograman berorientasi objek :

1. Objek (*Object*)

Objek adalah abtraksi dan sesuatu yang mewakili dunia nyata seperti benda, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi dan mempunyai operasi yang dapat diterapkan atau dapat berpengaruh pada status objeknya.

2. Kelas (*Class*)

   Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dalam kelas tersebut.

3. Pembungkusan (*Encapsulation*)

   Pembungkusan atribut data dan layanan yang mempunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

4. Pewarisan (*Inheritance*) dan Generalisasi/Spesialisasi

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dirinya.

5. Metode

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural.

6. *Polimorfisme*

   Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.2.4. *Unified Modelling Language* (UML)

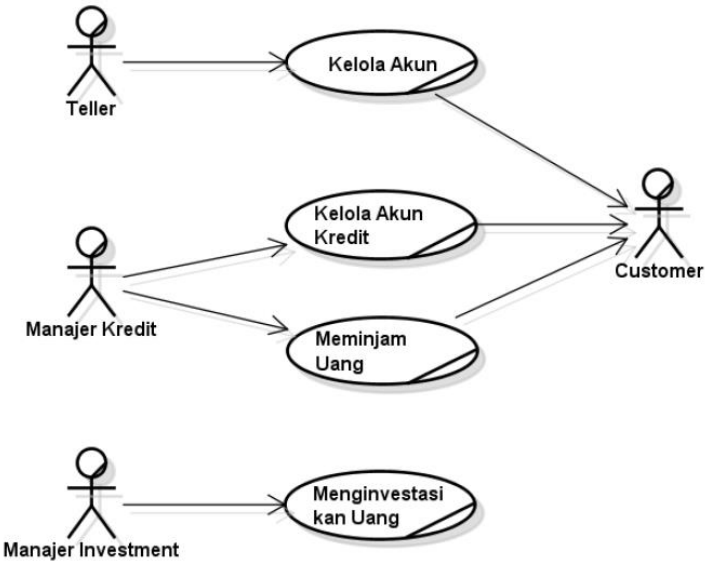
Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar untuk visualisasi, pemodelan, perancangan dan mendokumentasikan suatu sistem perangkat lunak berbasis objek yang terdiri dari kumpulan diagram, dikembangkan untuk membantu para pengembang sistem. Dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh dari *Rational Software* pada 1990-an. UML telah diadopsi oleh *Object Management Group* (OMG) pada tahun

2007 dan telah dikelola organisasi ini sejak saat itu. Pada tahun 2000 UML diterima oleh *International Organization for Standardization* (ISO) sebagai standar untuk pemodelan sistem perangkat lunak intensif.

Hingga saat ini, UML telah berkembang hingga versi 2.5. UML memiliki banyak sekali diagram, terutama versi terbaru UML yaitu UML 2.5. Berikut akan dijelaskan macam-macam diagram yang paling sering digunakan dalam pembangunan aplikasi berbasis objek [7].

1. *Business Use Case Diagram*

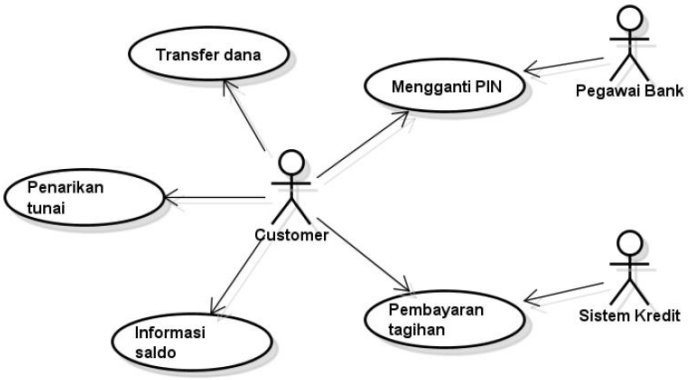
   Business use case diagram digunakan untuk mewakili fungsi yang disediakan oleh organisasi secara keseluruhan. *Business use case* juga digunakan secara luas selama kegiatan pemodelam bisnis untuk mengatur konteks sistem dan untuk membentuk sebuah fondasi untuk menciptakan *use case*.



Gambar 2. 2 Diagram *Business Use Case* untuk *Institusi Finansial* [7]

2. *Use Case Diagram*

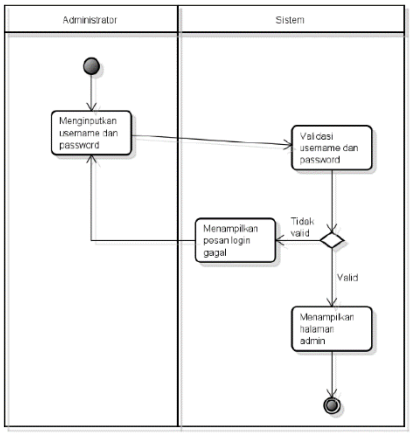
   Use case digunakan untuk memodelkan proses bisnis berdasarkan perspektif pengguna sistem. Use *case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.



Gambar 2. 3 *Use Case Diagram* untuk Sistem ATM [7]

3. *Activity Diagram*

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

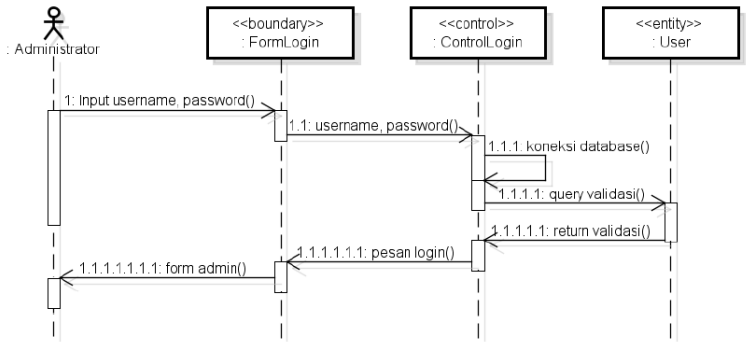


Gambar 2. 4 Diagram Aktifitas untuk Login Administrator [7]

4. *Sequence Diagram*

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*, diantaranya:

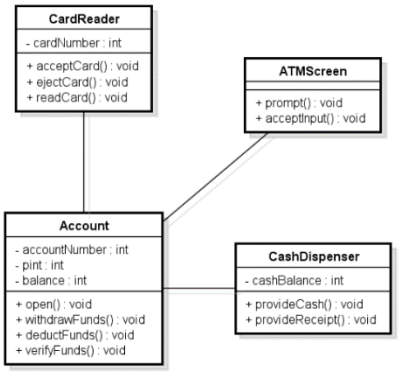
a. Interaksi yang terjadi antar *class*.
b. Operasi apa saja yang terlibat.
c. Urutan antar operasi.
d. Informasi yang diperlukan oleh masing-masing operasi.



Gambar 2. 5 *Sequence Diagram* untuk Proses Login [7]

5. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Selain itu, *class diagram* menunjukan juga atribut dan *method* yang ada pada setiap *class.*



Gambar 2. 6 *Class Diagram* untuk *Use Case* Penarikan Uang Sistem ATM [7]

2.2.5. PHP

PHP adalah singkatan dari PHP *Hypertext Preprocessor*. PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdrof.

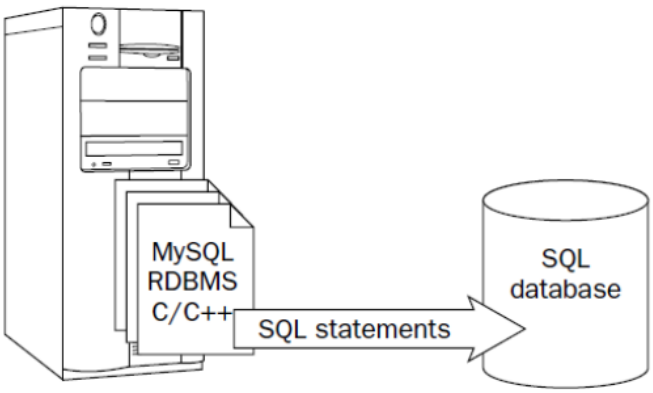
PHP merupakan software *Open-Source* yang disebarkan dan dilisensikan secara gratis serta dapat di *download* di situs resminya, http://www.php.net PHP ditulis dengan menggunakan bahasa C. Bahasa pemrograman PHP ini digunakan dalam proses pembangunan aplikasi sistem pembayaran menggunakan *radio frequency identification* (RFID) di *boarding school* pada SMP-SMA Terpadu Manahijul Huda.

Contoh *script*:

```
<?php
echo ("Hello Word");
?>
```

2.2.6. MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) bersifat *open source* yang memungkinkan data diakses dengan cepat oleh banyak pemakai secara bersamaan dan juga memungkinkan pembatasan akses pemakai berdasarkan *privilege* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (*structured query language*) yang merupakan bahasa standar pemograman database.



Gambar 2. 7 Interaksi MySQL RDMS dengan SQL

Beberapa perintah dasar SQL yang sering dipergunakan pada MySQL:

1. *Create Database*, perintah yang digunakan untuk membuat database baru.
   Sintaks : CREATE DATABASE DATABASE_NAME
2. *Drop Database,* perintah yang digunakan untuk menghapus database.
   Sintaks : DROP TABEL TABEL_NAME
3. *Create Tabel,* perintah yang digunakan untuk membuat tabel baru.
   Sintaks : CREATE TABEL tabel_name(CREATE_DEFINITION)
4. *Describe*, perintah yang digunakan untuk mendeskripsikan tabel.
   Sintaks : DESCRIBE (DESC) TABEL [COLUM]
5. *Alter Tabel,* perintah yang digunakan untuk memodifikasi tabel.
   Sintaks : ALTER [IGNOR] TABEL table_name
6. Drop *Tabel*, perintah yang digunakan untuk menghapus tabel.
   Sintaks : DROP TABEL tabel_name [tabel_name..]
7. *Delete*, perintah yang digunakan untuk menghapus record dri tabel.
   Sintaks : DELETE FROM tabel_name WHERE WHERE_DEFINITION
8. *Select*, perintah yang digunakan untuk query ke database.
   Sintaks : SELECT * FROM tabel_name

MySQL menyediakan banyak fitur yang mendukung sebuah lingkungan yang aman untuk menyimpan, memelihara, dan mengakses data. Berikut kelebihan-kelebihan yang ditemukan di MySQL:

1. Skalabilitas, MSQL dapat menangani database yang besar, yang telah dibuktikan implementasinya dalam organisasi seperti Yahoo, Google, Cisco, HP, NASA dan lain sebagainya.
2. Portabilitas, MySQL dapat berjalan pada berbagai macam sistem operasi termasuk Windows, Unix, Linux, Solaris dan Mac OS. Juga dapat

berjalan pada arsitektur yang berbeda, mulai dari low-end PC sampai high-end mainframe.

3. Konektivitas, MySQL sepenuhnya mendukung jaringan dan dapat diakses dari mana saja di internet serta pengguna dapat mengakses database MySQL secara bersamaan. MySQL juga menyediakan berbabagai macam API (Aplication Program interface) untuk mendukung konektivitas aplikasi yang ditulis dalam bahasa C, C++, Perl, PHP, Java, Phyton, C# dan lain sebagainya.
4. Keamanan, MySQL mencakup seluruh keamanan yang kuat untuk mengontrol akses data dan juga mendukung Secure Socket Layer (SSL) Protocol
5. Kecepatan, MySQL dikembangkan dengan kecepatan.
6. Mudah digunakan, MySQL mudah untuk digunakan dan diimplementasikan.
7. Open Source, MySQLAB membuat kode MySQL tersedia untuk digunakan setiap orang. filosofi open source memungkinkan khalayak global untuk berpartisipasi dalam pengembangan

2.2.7. Laravel

Dalam pengembangan web banyak sekali alat-alat yang digunakan untuk mempermudah pengembangannya. Salah satunya adalah dalam bentuk *framework.* Menurut Björemo & Trninić (2010), "*A framework in context of software development is a set of prewritten code or libraries which provide functionality common to a whole class of applications. The framework can be seen as a base or a skeleton to build upon*" [8]. Rosa & Shalahuddin (2013) mengatakan bahwa framework berupa kerangka kerja yang digunakan oleh programmer untuk mempermudahnya dalam membuat sebuah aplikasi dan mempermudah dalam melakukan perubahan [9]. Dapat disimpulkan bahwa framework merupakan kerangka kerja yang menyediakan fungsi – fungsi umum untuk programmer yang dapat digunakan untuk mempermudahnya dalam membuat sebuah aplikasi atau program

Menurut Björemo & Trninić (2010), keuntungan penggunaan framework adalah sebagai berikut [8]:

1. *Reuse Code*. Kode program yang telah dibuat, diuji, dan digunakan oleh programmer lain dapat digunakan/dikembangkan kembali sehingga dapat memangkas waktu dalam membuat program.
2. *Get free help*. Bantuan misalnya berupa adanya fungsi keamanan bawaan dari *framework* sehingga programmer secara tidak langsung mendapat bantuan secara gratis.
3. *Specific architecture or design pattern*. Membuat *programmer* mudah dalam memahami dan memulai membuat program dengan menggunakan *framework* dengan adanya konsep arsitektur pembuatan kode program pada framework.
4. *Support high level programming*. Terdapat beberapa modul dasar misalnya seperti login ataupun *database handling* yang dapat digunakan untuk mempercepat pengembangan program.
5. Terdapat *update* fitur dari *framework*.
6. Mudah dipahami oleh *new inexperienced developers*.

Ada berbagai jenis *framework* yang digunakan untuk pembuatan *software* berbasis *web*. Salah satu *framework* yang digunakan untuk pengembangan *software* berbasis *web* adalah Laravel. Laravel memiliki merupakan *framework* dengan *architecture pattern* yang berupa MVC atau *Model View Controller* [10]. Menurut Armel (2014), *developer* dapat menuliskan kode program dalam tiga basis berikut dalam konsep MVC:

1. *Model*

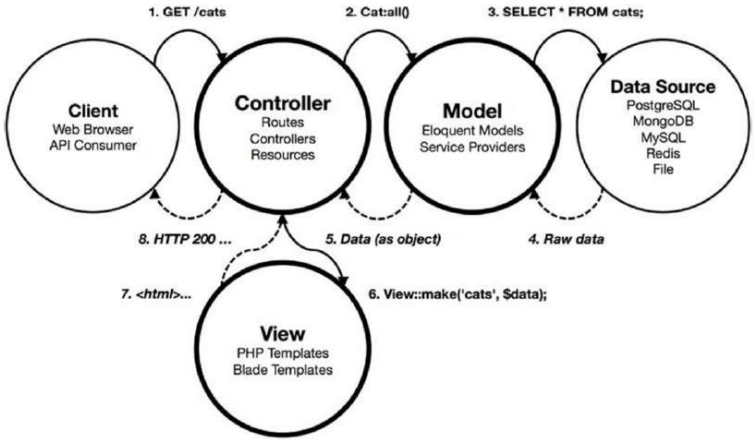
   Model merupakan bagian yang dapat digunakan untuk memanipulasi data. Model terdiri dari layer yang terletak antara data dan aplikasi.

2. *Views*

   Views merupakan gambaran visual untuk aplikasi web. Views bertanggungjawab untuk menampilkan data yang didapatkan oleh controller dari model.

3. *Controller*

   Fungsi utama dari controller adalah menangani requests dan menampaikan data dari model menuju ke views. Interaksi yang terjadi pada MVC dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Interaksi MVC Laravel

Keunggulan laravel dari *web* menurut Otwell (2014) adalah sebagai berikut [11]:

1. *RESTful Routing*

   Menggunakan cara yang mudah untuk merespon *requests* pada aplikasi.

2. *Command Your Data*

   Menggunakan Eloquent ORM dan sistem migrasi database yang dapat bekerja pada MySQL, Postgres, SQL Server, dan SQLite.

3. *Beautiful Templating*

   Dapat menggunakan native PHP atau blade template engine

4. *Ready For Tomorrow*

   Sempurna untuk semua ukuran kerja software seperti enterprise applications ataupun simple JSON APIs.

5. *Proven Foundation*

   Dibuat dengan Symfony components yang mana komponen tersebut well-tested and reliable code.

6. *Composer Powered*

   Composer merupakan alat yang digunakan untuk mengatur paket-paket pihak ketiga yang dapat diintegrasikan dengan Laravel.

7. *Great Community*

   Terdapat forum dan IRC Chat yang dapat digunakan untuk sharing dan bertanya.

8. *Red, Green, Refactor*

   Terdapat unit testing bawaan Laravel yang mudah digunakan.

Beberapa contoh *framework* PHP selain laravel adalah Symfony, Zend, Slim, Kohana, Lithium, dan CodeIgniter. Menurut Surguy (2014), Laravel dikembangkan berdasarkan fungsi yang belum ada pada beberapa *framework* tersebut dan beberapa *framework* tersebut masih fokus pada pengembangan untuk versi PHP yang lama (sebelum versi PHP 5.3). Berbeda dengan Laravel yang dikembangankan untuk versi PHP terbaru [12].

2.2.8. *Radio Frequency Identification* (RFID)

RFID (*Radio Frequency Identification*) adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau *transponder* (*tag*) untuk menyimpan dan mengambil data jarak jauh [13]. Label atau kartu RFID adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio [14].

RFID merupakan sebuah teknologi yang menggunakan media radio frekuensi dalam proses identifikasinya. Identifikasi dilakukan secara otomatis terhadap objek-objek atau manusia tanpa memerlukan operasi manual. Maka dapat disimpulkan, RFID adalah teknologi penangkapan data yang dapat digunakan secara elektronik untuk mengidentifikasi, melacak dan menyimpan informasi yang tersimpan dalam tag RFID.

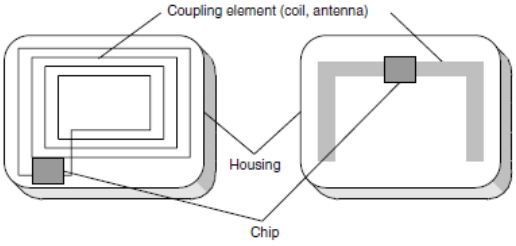
2.2.8.1.Komponen Komponen RFID

RFID secara umum terdiri atas tiga bagian. Ketiga bagian tersebut terdiri atas *tag*, *reader*, dan *host computer* [13] [14].

1. RFID *tag*, berisi antena sebagai coupling element yang memungkinkan untuk menerima dan merespon terhadap suatu *query* yang dipancarkan oleh suatu RFID *reader*. Juga terdapat *chip* yang mampu menyimpan sejumlah informasi unik.
2. *Reader* berfungsi untuk memudahkan di dalam membaca (*read)* informasi yang terkandung di dalam RFID *tag*. *Reader* sendiri terdiri atas terminal/RFID *reader mobile* (*reader tag* RFID), *Vehicle Mounted* RFID *reader* (dipasangkan pada kendaraan misalkan mobil untuk memudahkan di dalam pembacaan *tag* RFID), *Fixed* RFID *reader (*dipasangkan pada suatu tempat atau alokasi secara permanen, untuk kemudian *tag* RFID didekatkan kepada *fixed reader* ini).
3. *Host computer*, terdiri dari atas Basis data yang menyimpan semua data yang terkandung dalam *tag*. Sistem komputer yang mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara *tag* dan *reader*.

2.2.8.2.RFID Tag

RFID *tag* adalah sebuah benda kecil, komponen yang terdiri dari *chip* dan antenna. Tag ini dapat berupa stiker *adesif* yang ditempelkan pada suatu barang atau produk. Selain itu, *tag* ini juga dapat berupa koin dan kartu. Bentuk dan struktur dari *tag* ini dibuat fleksible sesuai dengan objek yang akan diidentifikasi. Gambar 2.9 menunjukkan layout dari sebuah RFID *tag*:



Gambar 2. 9 Layout RFID tag

2.2.8.3.RFID Reader

Prinsip kerja RFID *reader* serupa dengan *tranceiver* radio, yaitu memancarkan dan menerima. *Reader* ini dalam kondisi siaga akan memancarkan gelombang elektromagnetik sesuai dengan daya jangkaunya. Ketika ada *tag* memasuki area jangkauannnya, *tag* akan mendapat daya dari gelombang elektromagnet *reader* .

Dari daya yang diperoleh, *tag* memancarkan data yang dibawa. Data pancaran tersebut akan diterima oleh *reader* . Selanjutnya data yang diterima tadi akan diteruskan pada aplikasi untuk diolah sesuai dengan rancangan sistem.



Gambar 2. 10 Modul RFID *Reader* RC522

2.2.9. Arduino

Arduino adalah pengendali mikro single - board yang bersifat open - source, diturunkan dari wiring platform , dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardware dalam arduino memiliki prosesor Atmel AVR dan menggunakan software dan bahasa sendiri.

2.2.9.1.Sejarah Arduino

Proyek Arduino dimulai pertama kali di Ovre, Italy pada tahun 2005. Tujuan proyek ini awalnya untuk membuat peralatan control interaktif dan modul pembelajaran bagi siswa yang lebih murah dibandingkan dengan prototype yang lain. Pada tahun 2010 telah terjual dari 120 unit Arduino. Arduino yang berbasis open source melibatka n tim pengembang. Pendiri arduino itu Massimo Banzi dan

David Cuartielles, awalnya mereka memberi nama proyek itu dengan sebutan arduino dari ivrea tetapi seturut perkembangan zaman nama proyek itu diubah menjadi Arduino.

Arduino dikembangkan dari thesis hernando Barragan di desain interaksi institute Ivrea. Arduino dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor dan aktuator lainnya. Mikrokontroler pada board arduino di program dengan menggunkan bahasa pemrograman arduino 9 (*based on wiring*) dan IDE arduino (*based on processing*). Proyek arduino dapat berjalan sendiri atau juga bisa berkomunikasi dengan software yang berjalan pada komputer.

2.2.9.2.*Hardware* Arduino

Hardware dalam arduino memiliki beberapa jenis, yang mempunyai kelebihan dan kekurangan dalam setiap papannya. Penggunaan jenis arduino disesuaikan dengan kebutuhan, hal ini yang akan mempengaruhi dari jenis prosesor yang digunakan. Jika semakin kompleks perancangan dan program yang dibuat, maka harus sesuai pula jenis kontroler yang digunakan. Yang membedakan antara arduino yang satu dengan yang lainnya adalah penambahan fungsi dalam setiap papan circuitnya dan jenis mikrokontroler yang digunakan. Dalam tugas akhir ini, jenis arduino yang digunakan adalah arduino mega 2560, dengan Gambar sebagai berikut:



Gambar 2. 11 Arduino Mega 2560

Hardware arduino Mega 2560 memiliki spesifikasi sebagai berikut [15].

Table 2. 1 Speisifikasi Arduino Mega 2560

Mikrokontroler	Atmega2560
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Input Voltage (limit)	6-20
Pin Digital I/O	54 (yang 15 pin digunakan sebagai PWM *output*)
Pins Input Analog	16
Arus DC per pin I/O	20 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	256 KB (8 KB digunakan untuk *bootloader*)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHZ

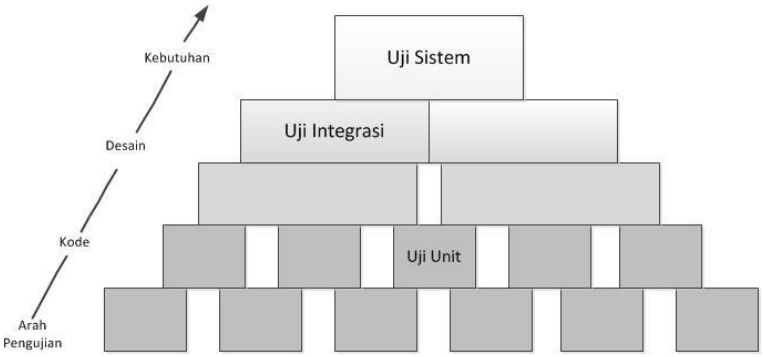
2.2.10. Pengujian

Pengujian adalah tahapan yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan dengan cara menemukan sebanyak-banyaknya kesalahan dalam selang waktu yang realistis.

Pengujian perangkat lunak sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak mengimplementasikan sebuah fungsi sfesifik dengan benar. Sedangkan mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pengguna.

2.2.10.1. Pengujian Verifikasi (*Verification*)

Pengujian verifikasi dilakukan mulai dari lingkup yang kecil naik ke lingkup yang besar seperti pada gambar 2.13 berikut ini:



Gambar 2. 12 Hirarki Pengujian Sistem

Pengujian diawali dari pengujian unit. Unit disini bisa berupa kumpulan kelas dan *package*. Setelah unit-unit selesai diuji maka dilakukan pengujian integrasi. Pengujian integrasi sebaiknya dilakukan secara bertahap, tidak dilakukan secara satu tahap langsung di akhir untuk menghindari kesulitan penelusuran jika terjadi kesalahan. Pengujian integrasi lebih kepada pengujian penggabungan dari dua atau lebih unit yang terdapat pada perangkat lunak. Setelah pengujian integrasi selanjutnya dilakukan pengujian sistem dimana unit-unit proses yang sudah diintegrasikan diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untuk menguji sistem perangkat lunak secara keseluruhan.

2.2.10.2. Pengujian Validasi (*Validation*)

Pengujian validasi memiliki beberapa pendekatan, diantaranya sebagai berikut:

1. *Black-Box Testing* (Pengujian Kotak Hitam)

   Pengujian kotak hitam adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah.

2. Pengujian Beta

   Pengujian Beta adalah pengujian yang dilakukan secara objektif dengan kata lain pengujian ini adalah pengujian secara langsung di lapangan atau tempat dimana aplikasi yang dibuat dimplementasikan.

3. Pengujian Perangkat Keras

   Pengujian Perangkat Keras adalah pengujian yang dilakukan ke pada alat yang sudah dirangkai atau di konfigurasi

2.2.11. Kuesioner

Mengingat terbatasnya masalah yang dapat ditanyakan dalam kuesioner, maka senantiasa perlu diingat agar pertanyaan-pertanyaan memang langsung berkaitan dengan hipotesis dan tujuan penelitian tersebut. Sebelum atau ketika membuat kuesioner, ada baiknya dipelajari kuesioner yang sudah ada dan relevan dengan topik penelitian yang akan dilakukan. (Singarimbun, M. Effendy, 1989)

Tujuan pokok pembuatan kuesioner penelitian adalah :

1. Memperoleh informasi yang relevan dengan tujuan survei.
2. memperoleh informasi dengan reliabilitas dan validitas setinggi mungkin.

<u>Isi Pertanyaan Kuesioner</u>

1. Pertanyaan tentang fakta. Umpamanya umur, pendidikan, agama, status perkawinan.
2. Pertanyaan tentang pendapat dan sikap. Ini menyangkut perasaan dan sikap responden tentang sesuatu.
3. Pertanyaan tentang informasi. Pertanyaan ini menyangkut apa yang diketahui oleh responden dan sejauh mana hal tersebut diketahuinya.

4. Pertanyaan tentang persepsi diri. Responden menilai perilakunya sendiri dalam hubungannya dengan yang lain.

Jenis Pertanyaan Kuesioner

1. Pertanyaan tertutup.
   Kemungkinan jawabannya sudah ditentukan terlebih dahulu dan responden tidak diberi kesempatan memberikan jawaban lain.
2. Pertanyaan terbuka.
   Kemungkinan jawabannya tidak ditentukan terlebih dahulu dan responden bebas memberikan jawaban.
3. Kombinasi tertutup dan terbuka.
   Jawabannya sudah ditentukan tetapi kemudian disusul dengan pertanyaan terbuka.
4. Pertanyaan semi terbuka.
   Jawabannya sudah tersusun tetapi masih ada kemungkinan tambahan jawaban.

Beberapa Cara Pemakaian Kuesioner

1. Kuesioner digunakan dalam wawancara tatap muka dengan responden. Cara ini yang lazim dilakukan
2. Kuesioner diisi sendiri oleh kelompok. Umpamanya. seluruh murid dalam satu kelas dijadikan responden dan mereka mengisi kuesioner secara serentak.
3. Wawancara melalui telepon. Cara ini sering dilakukan di Amerika Serikat dan negara-negara berkembang. Prosedur ini lebih mudah daripada wawancara tatap muka dan adakalanya orang tidak bersedia didatangi tapi bersedia diwawancara melalui telepon.
4. Kuesioner diposkan, dilampiri amplop yang dibubuhkan perangko, untuk dikembalikan oleh responden setelah diisi. Cara ini dapat dilakukan untuk kuesioner yang pendek dan mudah dijawab, tetapi mungkin cukup besar proporsi yang tidak dikembalikan oleh responden.