

BAB II

LANDASAN TEORI

2.1 Landasan Teori

Studi pustaka yaitu sekumpulan teori-teori yang menggambarkan konsep yang disusun secara sistematis dalam sebuah penelitian serta dijadikan bahan acuan dalam penelitian yang sedang dilakukan. Landasan teori didapatkan dari berbagai sumber yang dapat dipertanggung jawabkan kebenarannya, yaitu jurnal ilmiah, artikel ilmiah dan buku terkait.

2.1.1. Panti Asuhan

Panti asuhan adalah sebuah lembaga usaha kesejahteraan sosial yang memberikan pelayanan sosial kepada anak terlantar, yatim piatu serta memiliki tanggung jawab. Panti asuhan melakukan santunan serta pengetasan mental dan sosial pada anak asuh [2].

Panti asuhan merupakan suatu lembaga atau tempat yang menerima, menampung dan mengasuh anak yang ditinggalkan satu atau kedua orang tuanya, anak terlantar atau tidak mampu (dhuafa) sehingga para anak tersebut tetap dapat melanjutkan sekolah. Anak yatim piatu sangatlah penting kedudukannya khususnya di dalam ajaran agama Islam. Anak yatim dilarang untuk disisihkan, disakiti, diremehkan serta dirampas hak-haknya. Dalam agama Islam, yatim merupakan anak yang telah ditinggalkan orang tuanya dan harus disantuni, dimuliakan dan bahkan diperhatikan masa depannya[1].

2.1.2. Sistem Informasi

Syaifulloh dalam jurnalnya menyatakan menurut Robert A. Leitch dan K.Roscoe Davis (1983) mengatakan bahwa “Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan” [9].

Dalam jurnal lain yang ditulis oleh Ibrahim di tahun 2019 diuraikan beberapa pengertian dari sistem informasi menurut pakarnya[3], yaitu:

1. Andi ,2005.

Sistem informasi adalah suatu komponen yang saling berhubungan, saling mengumpulkan , memproses, menyimpan dan menyalurkan informasi untuk mendukung sebuah pengambilan keputusan serta kendali dalam sebuah organisasi atau lembaga.

2. Abdul kadir,2003.

Sistem informasi merupakan sebuah komponen yang terdiri dari manusia,komputer,teknologi informasi dan prosedur kerja yang memproses suatu data menjadi informasi dan dimasukan untuk mencapai suatu sasaran dan tujuan.

3. Steven Alter,2002.

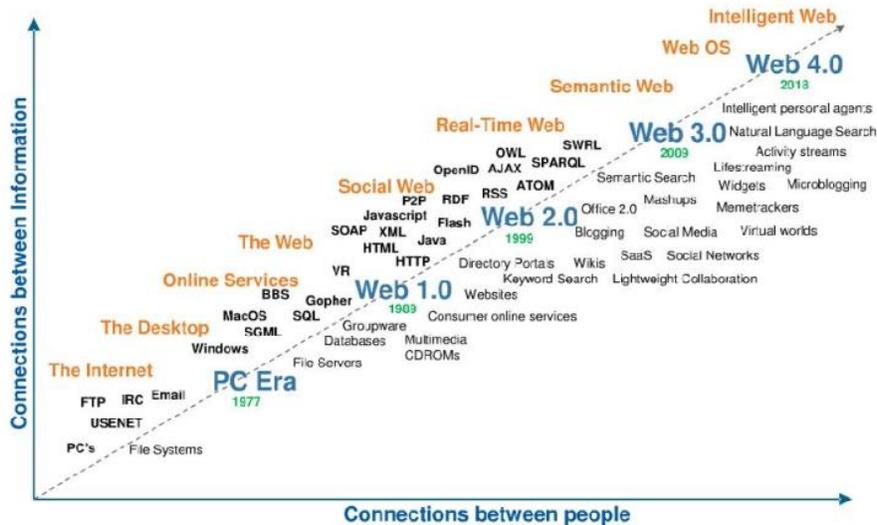
Sistem informasi adalah suatu sistem yang berada didalam suatu organisasi guna mendukung fungsi operasi organisasi dalam pengolahan transaksi harian yang bersifat managerial untuk menghasilkan laporan-laporan yang diperlukan.

2.1.3. Website

Website sebagai media penyampaian informasi memegang peranan yang sangat penting dalam era informasi digital. Melihat teknologi *website* yang jauh lebih maju dibandingkan dengan model teknologi yang digunakan pada aplikasi *desktop*. Hal ini sejalan dengan perkembangan model teknologi *website* dari masa ke masa yang mengalami perubahan. Perkembangan tersebut adalah sebagai berikut:

1. Pertama, model *website* statis (*web 1.0*) yang digagas dan ditemukan pertama kali oleh Tim Berners-Lee, dimana model *website 1.0* yang hanya mengutamakan penggunaan *script* atau kode HTML (*Hyper Text Markup Language*) dan juga sedikit menggunakan CSS (*Cascading Style Sheet*) dalam menampilkan isi konten yang statis pada halaman *website*. Isi konten yang ditampilkan hanya bisa dibaca atau dilihat oleh para pengunjung halaman *website*, tanpa adanya fitur atau fasilitas yang bisa digunakan untuk berkomunikasi secara langsung baik dengan pemilik *website* maupun para pengunjung lainnya.

2. Kedua, model *website open platform (web 2.0)* yang digagas oleh tim O'Reilly pada tahun 2005 merupakan sebuah model *website* yang bersifat baca – tulis (*read – write*) dimana para pengguna atau anggota dari sebuah *website* dapat berkontribusi untuk “meng-update” dan menyebarkan isi konten yang ada pada halaman *website* kepada anggota yang lainnya. Pada model *web 2.0*, halaman *website* yang ditampilkan menjadi lebih interaktif dan juga responsive. Hal tersebut dikarenakan proses komunikasi atau transfer data antara computer klien dengan server dilakukan secara *asynchronous*. Pada model *website 2.0* terdapat beberapa varian teknologi yang digunakan untuk proses pembuatannya yaitu penggunaan AJAX (*Asynchronous JavaScript and XML*), JSON (*JavaScript ObjectNotation*), DOM (*Document Object Model*), serta penggunaan Bahasa pemrograman *web* dinamis seperti JSP (*Java Server Pages*), ASP (*Active Server Pages*) dan PHP (*Hypertext Preprocessor*).
3. Ketiga, model *website semantic (web 3.0)* merupakan sebuah model *website* yang pertama kali dimunculkan dan diperkenalkan oleh tim Berners-Lee pada tahun 2001, yang kemudian dilanjutkan oleh John Markoff dari majalah New York Times pada tahun 2006. Model *web semantic* merupakan sebuah fase atau tahapan baru dalam perkembangan teknologi *web* yang akan digunakan di masa yang akan datang. *Model web 3.0* menggunakan Teknik kecerdasan buatan (*artificial intelligence*) yang digunakan untuk melakukan proses analisa dari penggunaan metadata yang digunakan dalam data mining terhadap pola hasil pencarian yang dilakukan oleh mesin tanpa adanya interfensi atau campur tangan manusia terhadap hasil pencarian yang ditampilkan pada halaman *website* (Wadhwa, 2015). Beberapa teknologi utama yang digunakan oleh *model web 3.0* adalah penggunaan *Resource Description Framework* (RDF), *RDF Schema* (RDFS), bahasa query untuk RDF yaitu SPARQL dan *Simple Knowledge Organization System*.
4. Keempat, *Model Website Symbiotic* atau *Ubiquitous Pragmatic (web 4.0)* merupakan *model website* yang dapat dikatakan sebagai *role model* bagi teknologi *website* di masa yang akan datang. Belum banyak hal yang dapat disampaikan dan ditulis mengenai *model website* ini. Hal tersebut dikarenakan masih terbatasnya sumber atau bahan literatur yang membahas secara jelas dan spesifik mengenai *model website 4.0*.



Gambar II. 1 Model Web 1.0 sampai Model 4.0

Sumber : (Himawan, 2017).

Website adalah kumpulan halaman web yang saling terkait dan biasanya disimpan di dalam komputer web server

2.1.4. Web Service

Web service adalah suatu fasilitas yang digunakan untuk melayani aplikasi web yang akan kita buat. web browser seperti explorer atau navigator berkomunikasi melalui jaringan (termasuk jaringan internet) dengan web server, menggunakan HTTP. Browser akan mengirimkan request kepada server untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh server. Server memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protoko HTTP

Menurut Eko Budi setiawan dan Angga Try Ramdani dalam bukunya yang berjudul “Membangun Aplikasi Android, Web dan Web Service” mengatakan bahwa kelebihan dari web service yaitu [23]:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup, sehingga segi keamanan dapat dikendalikan.

2. *Web service* memiliki kemudahan dalam *deployment*, karena tidak memerlukan *registrasi*. *Web service* cukup di-*upload* ke *web server* dan dapat diakses oleh pihak-pihak yang memiliki otorisasi.
3. *Web service* dijalankan di *port* 80 yang merupakan standar protokol bagi HTTP, sehingga *web service* tidak memerlukan konfigurasi khusus dari sisi *firewall*.

2.1.5. Aplikasi Berbasis Website

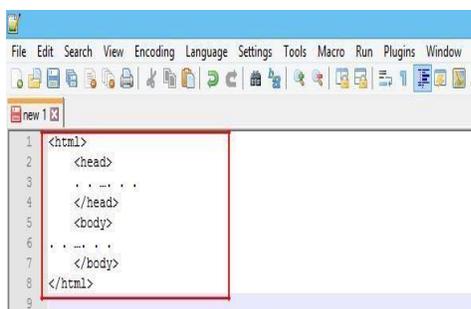
Aplikasi berbasis *website* adalah aplikasi sisi *server* mengatur informasi yang akan dikirimkan ke *web browser* dan memproses *form* yang diterima dari *web browser*. Meskipun terdapat sedikit kelemahan, menimbang keuntungan yang diperoleh, pengembangan aplikasi otomatis berbasis *web* layak untuk dikembangkan lebih lanjut. Hal ini ditunjang juga dengan banyaknya modul– modul dan alat bantu untuk pengembangannya.

Web juga merupakan suatu kumpulan *hyperlink* yang menuju alamat satu ke alamat lainnya dengan Bahasa HTML (*HyperText Markup Language*). *Website* dapat diartikan sebagai kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait.

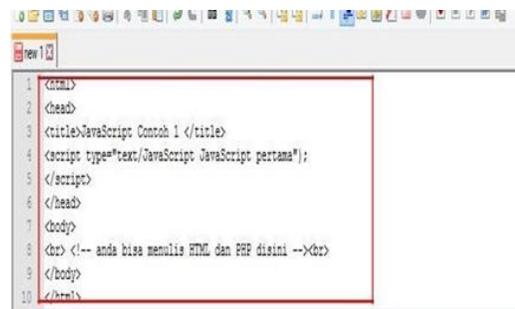
2.1.6. HTML

HTML (*Hyper Text Markup Language*) adalah sekumpulan simbol–simbol atau tag–tag yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. Tag – tag HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana x tag HTML itu seperti b, i, u, dan lainnya [24].

HTML adalah Bahasa terstandarisasi yang digunakan oleh *browser* untuk menciptakan antarmuka pengguna. HTML adalah Bahasa pemrograman yang digunakan sebagai (*formatting tools*). Format adalah mekanisme yang mengatur bentuk *visual* yang tampil pada *web browser* seperti *netscape navigator* atau *internet explorer*. Objek yang dapat diformat antara lain adalah teks, *style* dari teks (*font*), ukuran, tebal, gambar, suara, koneksi (*link*) dan lainnya.



```
1 <html>
2 <head>
3   . . . . .
4 </head>
5 <body>
6   . . . . .
7 </body>
8 </html>
9
```



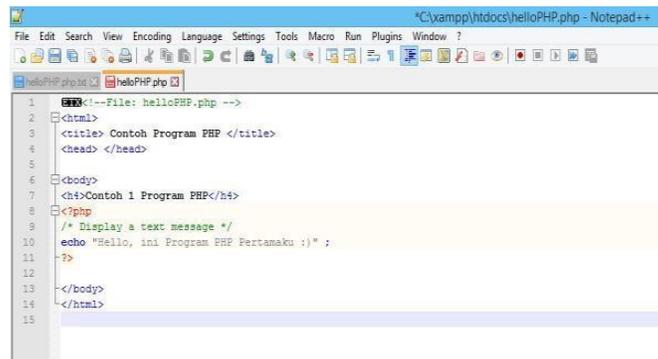
```
1 <html>
2 <head>
3 <title>JavaScript Contoh 1 </title>
4 <script type="text/JavaScript JavaScript pertama">
5 </script>
6 </head>
7 <body>
8 <br> <!-- anda bisa menulis HTML dan PHP disini --><br>
9 </body>
10 </html>
```

Gambar II. 2 Contoh Kodingan Format HTML

2.1.7. PHP

Hypertext Preprocessor merupakan Bahasa *script web HTML embedded* dengan menyisipkan kode PHP dalam HTML. Saat kode PHP dibaca oleh server, maka *output* dari fungsi PHP pada halaman biasanya dikembalikan menjadi kode HTML sebelum halaman *web* dibuka dan pengguna tidak bisa melihat kode PHP pada halaman itu. Tujuan utama dari Bahasa PHP adalah memungkinkan perancang *web* untuk menuliskan halaman *web* dinamik secara cepat. Sistem *database* yang didukung oleh PHP adalah: *oracle, MySQL, Ms.Access, Sybase* [23]

Gambar II. 3 Contoh format Kodingan PHP



```
1  <!--File: helloPHP.php -->
2  <html>
3  <title> Contoh Program PHP </title>
4  <head> </head>
5
6  <body>
7  <h4>Contoh 1 Program PHP</h4>
8  <?php
9  /* Display a text message */
10 echo "Hello, ini Program PHP Pertama! :)" ;
11 ?>
12
13 </body>
14 </html>
15
```

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP mengizinkan pengembangan untuk menempelkan kode didalam HTML dengan menggunakan Bahasa yang sama, seperti *Perl* dan *UNIX Shells*.

PHP adalah suatu Bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh computer yang bersifat *server – side* yang dapat ditambahkan ke dalam HTML.

PHP adalah skrip bersifat *server side* yang ditambahkan ke dalam HTML. Skrip ini akan membuat suatu aplikasi dapat terintegrasi ke dalam HTML sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi bersifat dinamis.

2.1.8. CSS

CSS adalah singkatan dari *Cascading Style Sheet*. CSS biasanya selalu dikaitkan dengan HTML, karena keduanya memang saling melengkapi dimana HTML ditujukan untuk membuat struktur, atau konten dari halaman *web*. Sedangkan CSS digunakan untuk tampilan dari halaman *web* tersebut. Istilahnya “*HTML for Content, CSS for Presentation*” [23].

2.1.9. Bootstrap

Bootstrap merupakan *framework* CSS yang digunakan untuk membantu membuat *website responsive* dengan mudah. *Website* yang dibuat dengan *bootstrap* desainnya akan menyesuaikan dengan ukuran *layer device* yang digunakan sehingga tampilan pada *smartphone* berbeda dengan tampilan pada laptop, namun tetap terlihat rapi [12].

2.1.10. Definisi Xampp

Menurut Kartini (2013:26)[25] , “XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket, adapun paketnya sudah terdapat *Apache (Web Server)*, *MySQL (Database)* *PHP (Server Side Scripting)*, *Perl*, *FTP Server*, *PhpMyadmin* dan berbagai pustaka bantu lainnya. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server apache*.

Adapun pengertian dari XAMPP adalah sebagai berikut :

1. X yang berarti program ini dapat dijalankan di banyak sistem operasi.
2. A yang berarti *Apache* merupakan suatu aplikasi *web server*.
3. M yang berarti *MySQL* merupakan untuk penyimpanan *database server*.
4. P yang berarti *PHP* merupakan bahasa pemrograman yang dipakai untuk membangun *website* dinamis.
5. P yang berarti *Perl* bahasa pemrograman untuk segala keperluan. *Perl* merupakan penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum, perl sangat populer digunakan dalam pemrograman CGI (*Common Gateway Interface*).

2.1.11. Database MySql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*), *database* ini *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus yang bersifat khusus.

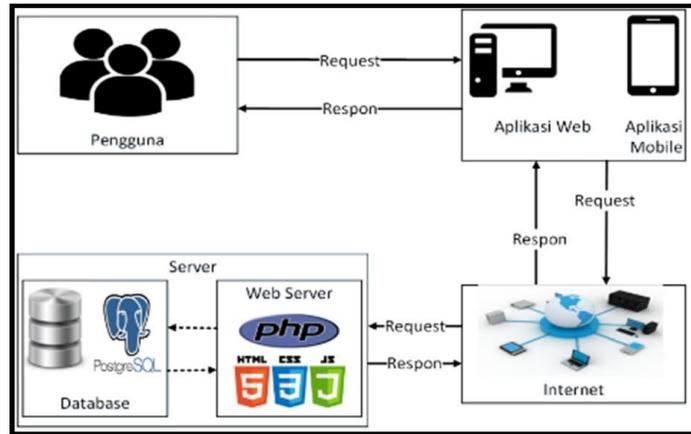
Kekuatan MySQL tidak ditopang oleh sebuah komunitas, seperti *Apache*, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing-masing, tetapi MySQL didukung penuh oleh sebuah perusahaan profesional dan komersial, yakni MySQL AB dari Swedia.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana 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 mudah secara otomatis [23].

Menurut madcoms, MySql adalah sistem manajemen database SQL yang bersifat open source dan paling populer saat ini. Sistem database MySql mendukung beberapa fitur seperti multithreaded. Multi-user dan SQL DBMS (*data base Management System*)[22]

2.1.12. Arsitektur Sistem

Arsitektur sistem menggambarkan bagaimana jalur komunikasi pada software. Arsitektur sistem palikasi pada website dapat dijelaskan melalui gambar berikut :



Gambar II. 4 Contoh alur Arsitektur Sistem

Sebagai berikut penjelasan pada gambar diatas yaitu:

1. Pengguna *website* melakukan *request* data ke *web server* melalui *internet*.
2. *Internet* meneruskan *request* dari pengguna *website* ke *web server*.
3. *Web server* menerima *request* dari pengguna *website*, setelah itu *web server* melakukan *request* data di *database* sesuai *request* pengguna *website*.
4. *Database* menerima *request* dan me-*response* ke *web server* sesuai *request* pengguna *website*.
5. *Web server* mengirim data sesuai dengan *request* pengguna *website* melalui *internet*.
6. *Internet* meneruskan *respon* dari *web server* ke pengguna / *user*.

2.1.13. UML

Menurut Nugroho dalam Melani & Bayu , mengemukakan bahwa ”UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek)[26].”



Gambar II. 5 Logo UML

Tujuan Penggunaan UML (Unified Modeling Language)

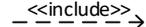
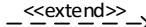
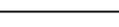
1. Memberikan model yang siap pakai, bahasa permodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa permodelan yang bebas dari berbagai bahasa Pemrograman dan proses rekayasa.

A. Use case Diagram

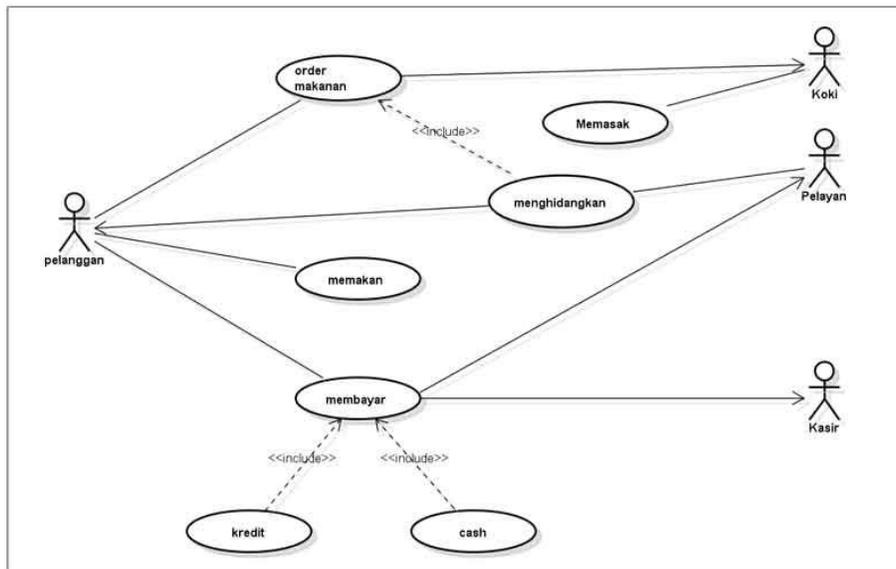
Use case Diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. *Use case diagram* digunakan selama proses analisi untuk mengetahui kebutuhan sistem dan untuk memahami bagaimana sistem harusnya bekerja[a]. Berikut simbol-simbol *use case diagram* beserta contoh dari *use case diagram*:

Tabel II. 1 Simbol-simbol Use Case Diagram

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 eksplisit.
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 <i>actor</i> .
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

Sebagai berikut contoh penerapan dari simbol-simbol *usecase diagram* diatas



Gambar II. 6 Contoh Use Case Diagram

B. Activity Diagram

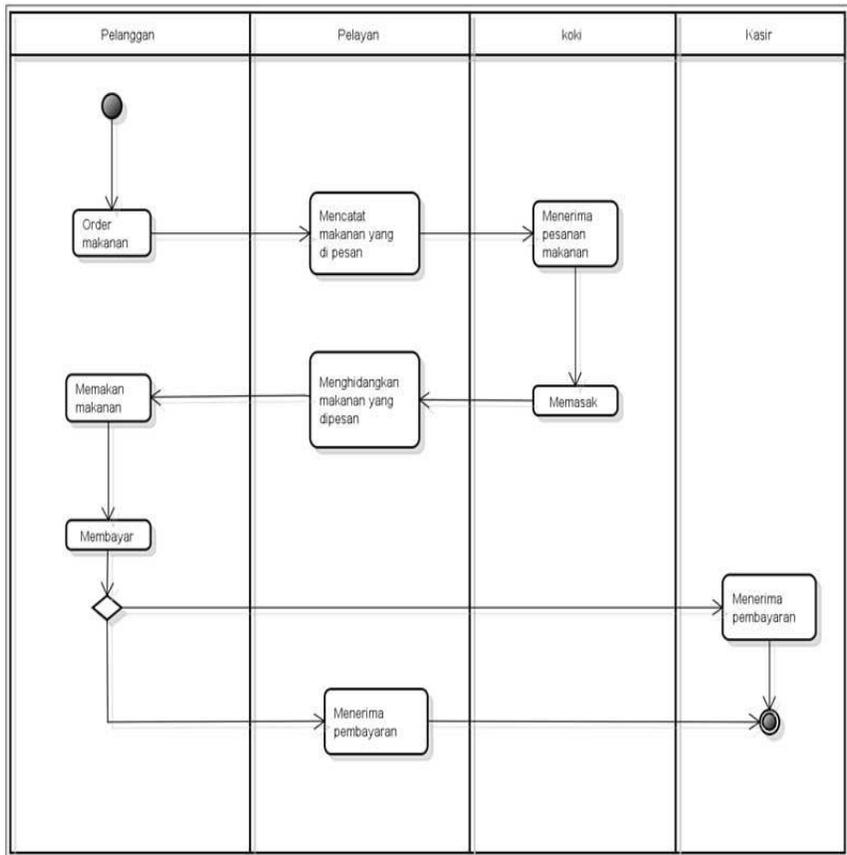
Activity Diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa esksekusi. Sebuah aktivitas dapat direalisasikan oleh satu *usecase* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *usecase* menggambarkan bagaimana aktor menggunakan sistem untuk aktifitas tertentu. Sebagai berikut adalah simbol-simbol *Activity diagram*:

Tabel II. 2 Simbol-simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN

1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

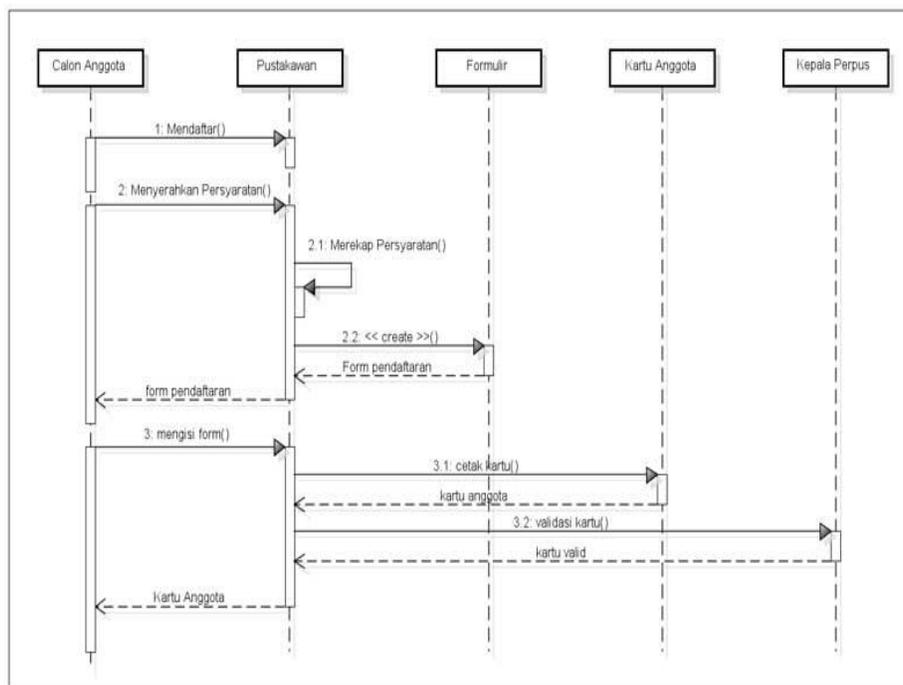
Berikut contoh penerapan dari simbol-simbol *Activity diagram* diatas :



Gambar II. 7 Contoh Activity Dagram

C. Sequence Diagram

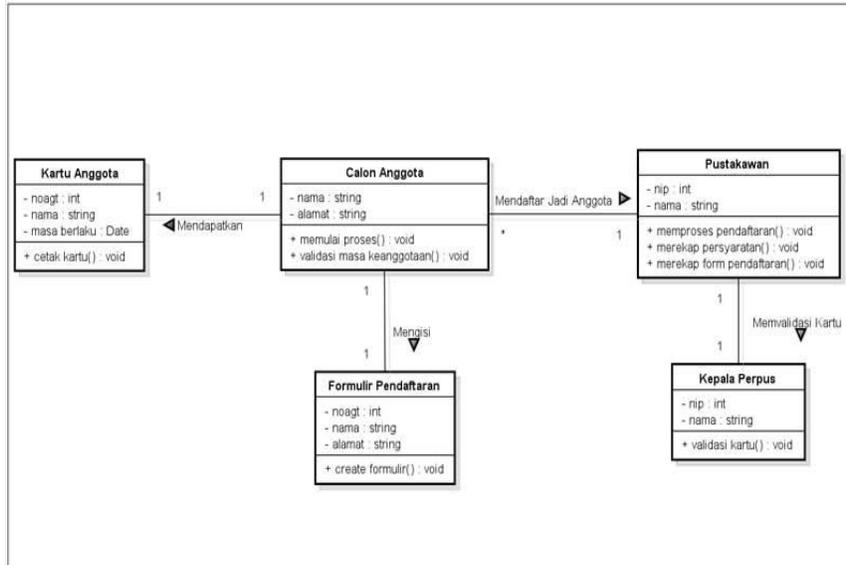
Sequence Diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*



Gambar II. 8 Contoh *sequence Diagram*

D. Class Diagram

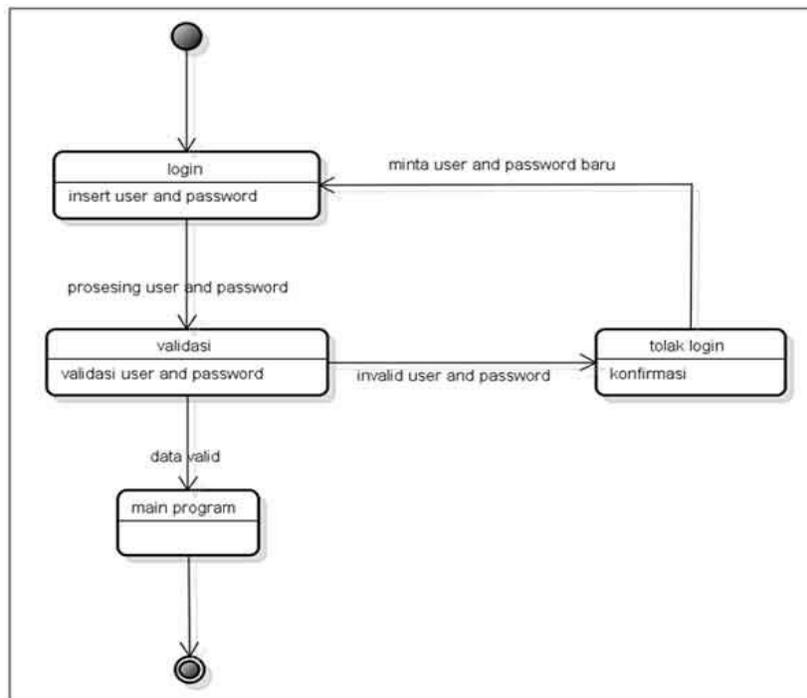
Class Diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi *diagram* ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.



Gambar II. 9 Contoh Class Diagram

E. State Machine Diagram

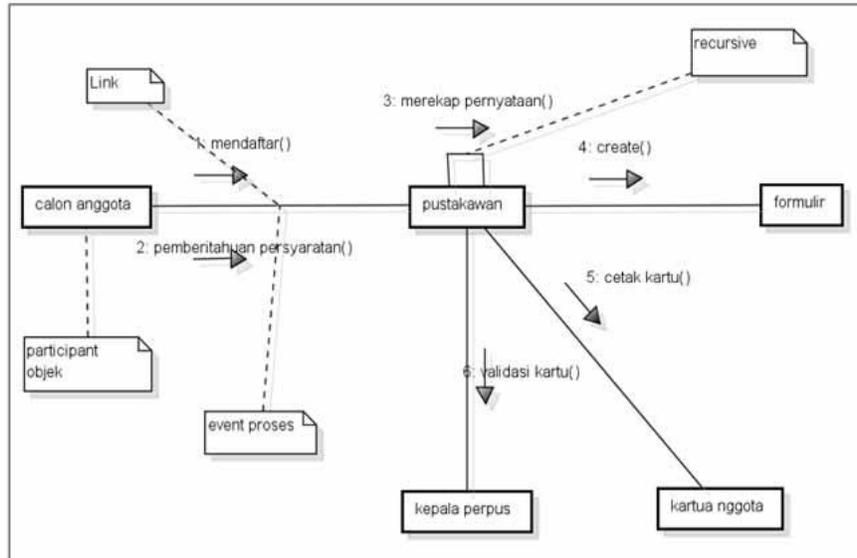
State Machine Diagram yaitu salah satu jenis diagram pada UML yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.



Gambar II. 10 contoh dari State Machine Diagram

F. Communication Diagram

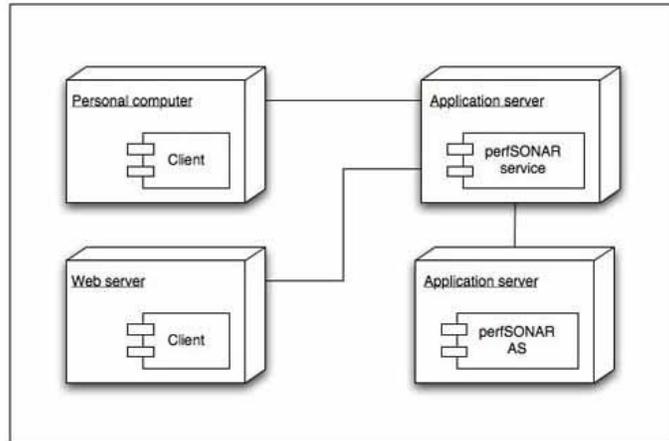
Communication Diagram yaitu salah satu jenis diagram pada UML yang dapat menggambarkan tahapan terjadinya suatu aktivitas dan *diagram* ini juga menggambarkan interaksi antara objek yang ada pada sistem. Hampir sama seperti *sequence diagram* akan tetapi *communication diagram* lebih menekankan kepada peranan masing-masing objek pada sistem.



Gambar II. 11 Contoh *Communication Diagram*

G. *Deployment Diagram*

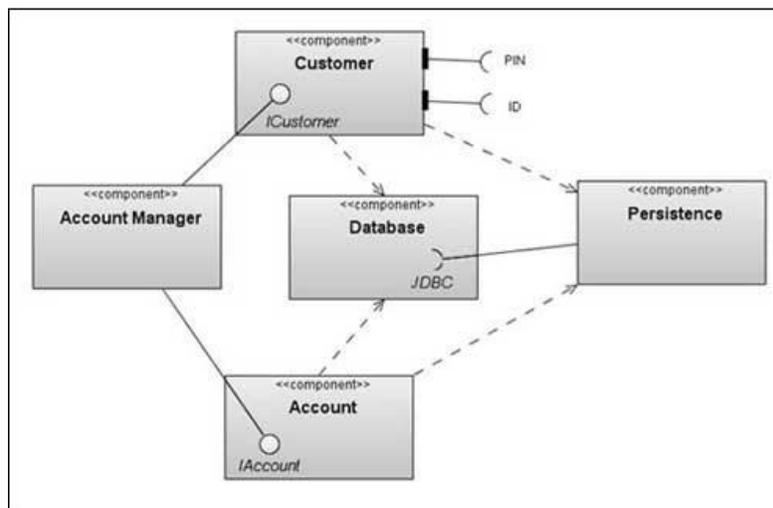
Deployment diagram yaitu salah satu diagram pada UML yang menunjukkan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. Jadi *Deployment diagram* intinya untuk menunjukkan letak *software* pada *hardware* yang digunakan sistem.



Gambar II. 12 Contoh *Deployment Diagram*

H. Component diagram

Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa *file data* atau *.exe*, *source code*, *table*, dokumen dsb.



Gambar II. 13 contoh dari *Component Diagram*.

I. Object Diagram

Object diagram yaitu salah satu jenis diagram pada UML yang menggambarkan objek-objek pada suatu sistem dan hubungan antarnya.

J. Composite Structure Diagram

Composite structure diagram yaitu salah satu jenis diagram pada UML yang menggambarkan struktur internal dari penklasifikasian (*class, component atau use case*) dan termasuk titik-titik interaksi penklasifikasian sebagian lainnya dari suatu sistem. Ini hampir mirip seperti class diagram akan tetapi *composite structure diagram* menggambarkan bagian-bagian dari individu kelas saja bukan semua kelas.

K. Interaction Overview Diagram

Interaction Overview diagram yaitu salah satu jenis diagram pada UML yang berguna untuk men-visualisasikan kerjasama dan hubungan antara activity diagram dengan sequence diagram.

L. Package diagram

Package Diagram yaitu salah satu jenis diagram pada UML digunakan untuk mengelompokkan kelas dan juga menunjukkan bagaimana *elemen model* akan disusun serta menggambarkan ketergantungan antara paket-paket.

M. Diagram Timing

Diagram timing yaitu salah satu jenis diagram pada UML yang disebut sebagai bentuk lain dari interaksi diagram, dimana fokus yang paling utamanya kepada waktu. *Diagram timing* berguna untuk menunjukkan faktor-faktor yang membatasi waktu antara perubahan state terhadap objek yang berbeda.

2.1.14. Black Box Testing

Blackbox testing adalah metode untuk pengetesan sebuah aplikasi. *Blackbox testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineers untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional sebuah program[27]. *Blackbox testing* memiliki dua teknik pengujian yaitu:

1. *Acceptance testing* adalah pengujian terakhir sebelum sistem dipakai oleh *user* yang melibatkan pengujian dengan data dari pengguna sistem.
2. *User acceptance test* adalah uji terima perangkat lunak yang dilakukan ditempat pengguna oleh *user* perangkat lunak tersebut. Tujuan pengujian ini adalah untuk menguji apakah sistem sudah sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem (validasi).

2.2. Tinjauan Organisasi

2.2.1. Sejarah Organisasi

Panti Asuhan Muhammadiyah Bandung berdiri dengan dilatar belakangi oleh kondisi sosial masyarakat sekitar Kelurahan kebon pisang kecamatan Sumur Bandung yang berada di daerah padat penduduk dan banyak anak anak yatim, piatu dan dhuafa yang putus sekolah karena terjadinya krisis moneter pada tahun 1996 sehingga banyak orang tua yang di PHK sehingga tidak mampu untuk membiayai sekolah anak anaknya. Banyak anak anak yg putus sekolah terutama anak anak yatim piatu di daerah sekitar sumur bandung, disamping itu juga dg adanya program kristenisasi di daerah cimahi, cianjur selatan dan ranca buaya maka dari itu untuk menanggulangi masalah tersebut maka kami Pimpinan Cabang Muhammadiyah Sumur Bandung mendirikan panti asuhan pada tanggal 17 Juli 1997, bertempat di Jl. Veteran No. 118 dengan Kepala Panti nya Bapak Arif Nurdin Sekretaris Bapak Ngadiono, Bendahara Dedi Junaedi dan Wali Asrama Suhud Suparno dengan jumlah anak asuh sebanyak 21 orang.

Pada Tahun 1999 Membeli rumah di Jl. Babakan Ciamis untuk Asrama Putri karena untuk kenyamanan anak asuh agar terpisah antara asrama putra dan putri. Pada Tahun 2013 Membeli Rumah di Jl. A.H Nasution 140 Sukamiskin untuk asrama putra sehubungan dengan makin banyaknya anak asuh yang dibina. Pada Tahun 2014 Penambahan asrama Putri di Sindanglaya dengan jumlah anak asuhnya 10 orang putri. Pada Tahun 2018 jumlah anak asuh yang dibina sebanyak 76 orang.

Nama kepala panti :

1. Arif Nurdin (1997- 1999).
 2. M. Amin Rahim (1999 - 2002).
 3. Riyana Sutisna (2003 – 2006).
 4. Solehudin (2007 – 2011).
 5. Solehudin (2012 – 2016).
- Udin Nasrudin (2017 – 2021)

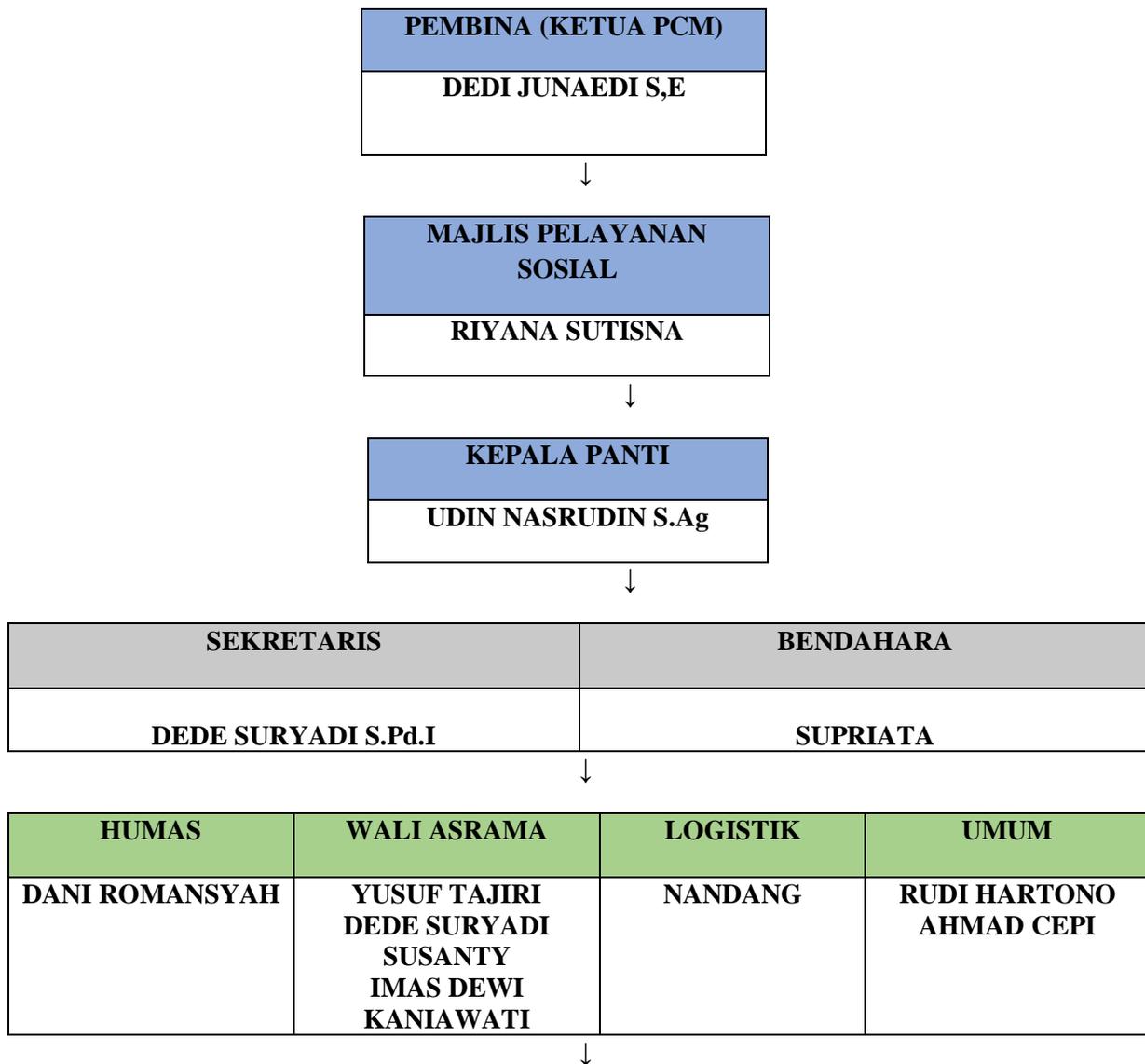
2.2.2. Logo Organisasi

Berikut logo dari panti Asuhan Muhammadiyah Bandung :



2.2.3. Struktur Organisasi

STRUKTUR ORGANISASI PANTI ASUHAN MUHAMMADIYAH SUMUR BANDUNG



ANAK ASUH			
ASRAMA VETERAN 15 ANAK	BABAKAN CIAMIS 30 ANAK	A.H NASUTION 21 ANAK	SINDANG LAYA 10 ANAK

2.2.4. Tugas dan Tanggung Jawab

Berikut rincian tugas dan tanggung jawab pengelola:

A. Tugas Pembina (Ketua PCM Sumur Bandung)

1. Bertanggung jawab atas keberlangsungan Panti Asuhan Muhammadiyah Sumur Bandung.
2. Membina dan mengawasi jalannya panti asuhan.
3. Memberikan solusi terhadap permasalahan yang terjadi apabila tidak dapat diselesaikan oleh MPS.

B. Tugas MPS (Majelis Pelayanan Sosial)

1. Bertanggung jawab kepada Pembina Panti asuhan (PCM)
2. Membuat program untuk meningkatkan kesejahteraan Panti Asuhan
3. Mengawasi operasional Panti Asuhan

C. Tugas Kepala Panti

1. Bertanggungjawab kepada MPS sebagai atasan.
2. Menyusun program kerja panti asuhan.
3. Melaksanakan program kerja Panti asuhan.
4. mengevaluasi program yang telah di buat.
5. Melaporkan kegiatan Panti Asuhan kepada MPS.

D. Tugas Sekretaris

1. Membantu kepala panti dalam melaksanakan pekerjaan teknis.
2. Mencatat administrasi Panti asuhan meliputi surat menyurat.
2. Membuat laporan Laporan kegiatan Panti Asuhan.

E. Tugas Bendahara

1. Menerima, menyimpan, membayarkan, menatausahakan dan mempertanggung jawabkan uang untuk keperluan belanja yang diketahui oleh Kepala Panti Asuhan .
2. Meneliti kelengkapan perintah pembayaran yang diterbitkan oleh Kepala Panti asuhan.
3. Menguji kebenaran perhitungan tagihan yang tercantum dalam perintah pembayaran.
4. Menguji ketersediaan dana yang bersangkutan.
5. Menyediakan uang persediaan dan merencanakan penarikan dana sesuai keperluan belanja Panti.
6. Melaksanakan penatausahaan dan pengarsipan surat kedinasan, dokumen keuangan lainnya.
7. Melaksanakan pembukuan sesuai peraturan yang berlaku.

F. Tugas HUMAS

1. Menjalin hubungan baik dengan masyarakat sekitar Panti Asuhan.
2. Mempromosikan Panti Asuhan kepada Masyarakat.
3. Meneliti kondisi obyektif anak asuh yang akan tinggal di Panti Asuhan.

G. Tugas Seksi Logistik

1. Menyediakan kebutuhan peralatan makanan, kesehatan dan sekolah anak.
2. Menginventarisir peralatan yang ada di panti asuhan.

H. Tugas Pembantu Umum

1. Membantu dalam pelaksanaan program panti asuhan.
2. Menjalin kerjasama dengan pengurus yang lain.

I. Tugas Dan Tanggung Jawab Wali Asrama

1. Membimbing dan mengasuh anak sesuai dengan usia dan kebutuhannya.
 2. Menyiapkan kebutuhan sehari hari anak asuh makan dan pakaiannya.
 3. Menyiapkan kebutuhan sekolah anak asuh.
 4. Mengawasi perilaku anak selama berada di lingkungan panti asuhan.
 5. Memantau perkembangan pendidikan anak asuh.
- Melaporkan kegiatan kepada Kepala Panti Asuhan.

