BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Adapun beberapa landasan teori yang dipakai dalam penyusunan Laporan Sistem Monitoring Perkembangan Santri Madrasah Mardliyah diantaranya:

2.1.1 Konsep Dasar Sistem

Istilah sistem bukanlah hal yang asing bagi kebanyakan orang. Seringkali sistem mengacu pada komputer seperti IBM PC atau Macintosh, tetapi juga bisa ke arah yang lebih luas seperti sistem tatasurya atau bahkan ke hal-hal yang lebih spesifik seperti sistem respirasi mamalia.

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudnakn untuk mencapai suatu tujuan [2]. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem.

2.1.1.1 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu :

1. Tujuan

   Setiap sistem memiliki tujuan, entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali.

2. Masukan (*Input*)

   Masukan sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak sejara fisil) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkah contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa dari pelanggan).

3. Proses

   Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran (*Output*)

   Keluaran merupakan hasil dari pemrosesan.

5. Mekanisme Pengendalian dan Umpan Balik

   Mekanisme pengendalian diwujudkan dengan menggunakan umpan balikm yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan. Dalam bentuk yang sederhana, dilakukan perbandingan antara keluaran sistem dan keluaran yang dikehendaki (standar). Jika terdapat penyimpangan terletak pada proses, maka prosesnyalah yang diperbaiki.

6. Batas

   Yang disebut batas sistem adalah pemisah antara sistem dan daerah diluar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelomtong dipengaruhi oleh pembelian oleh pelanggan, gerakan pesaing, dan ketersediaan dana dari bank. Tentu saja, batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbatasan dana.

7. Lingkungan

   Lingkungan adalah segala sesuatu yang berada di luar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang

merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem. Lingkungan bagi sebuah organisasi dapat berupa vendor, pelanggan, pemilik, pemerintah, bank, dan bahkan pesaing.

2.1.2 Sistem Informasi

Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS). Dalam prakteknya, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang paling penting.

Ada beragam definisi informasi, sebagaimana tercantum pada tabel 2.1. dari berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan [2].

Tabel 2.1 Pengertian Sistem Informasi

Sumber	Definisi
Alter (1992)	Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
Bodnar dan Hopwood (1993)	Sistem informasi adalah kumpulan perangkat keras dan oerangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
Gelinas, Oram, dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum

Sumber	Definisi
	terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada pemakai.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
Turban, McLean, dan Weterbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
Wikinson (1992)	Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sararan perusahaan. (A.S. and Shalahuddin, 2011)

2.1.2.1 Komponen Sistem Informasi

Dalam suatu sistem informasi terdapat komponen-komponen seperti :

1. Perangkat keras (*hardware*)

   Mencakup peranti-peranti fisik seperti komputer dan printer.

2. Perangkat lunak (*software*) atau program

   Sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.

3. Prosedur

   Sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang

   Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

5. Basis data (*database*)

   Sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.

6. Jaringan komputer dan komunikasi data

   Sistem penghubung yang memungkinkan sumber dipakai ecara bersama atau diakses oleh sejumlah pemakai.

Pada prakteknya, tidak semua sistem informasi mencakup keseluruhan komponen-komponen tersebut. Sebagai contoh, sistem informasi pribadi yang hanya melibatkan sebuah pemakai dan sebuah komputer tidak melibatkan fasilitas jaringan dan komunikasi. Namun, sistem informasi grup kerja yang melibatkan sejumlah orang dan sejumlah komputer, memerlukan sarana jaringan dan komunikasi.

2.1.3 Monitoring

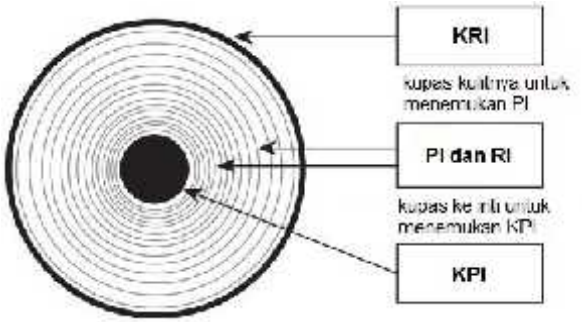
Monitoring atau pemantauan adalah suatu seni mengumpulkan informasi dengan usaha yang minimum untuk membuat suatu keputusan di waktu yang tepat [Gudda, 2011]. Informasi yang sudah dikumpulkan dapat digunakan untuk kemudian dilakukan analisis, diskusi, maupun pembuatan laporan. Monitoring memungkinkan kita untuk mematau proses secara rutin dalam jangka waktu tertentu, menentukan apakah suatu proses tersebut sudah berjalan dengan baik atau sudah sesuai dengan tujuan awal.

2.1.3.1 Key Result Indicator

Terdapat empat tipe ukuran kinerja [Parmenter, 2001] :

1. Key Result Indicator (KRI) memberitahu bagaimana kinerja anda dari satu perspektif atau faktor keberhasilan kritis
2. Result Indicator (RI) memberitahu apa yang telah anda lakukan.
3. Performance Indicator (PI) memberitahu anda apa yang harus dilakukan
4. KPI memberitahu anda apa yang harus dilakukan untuk meningkatkan kinerja secara dramatis.

Hubungan dari keempat ukuran ini dianalogikan dengan sebuah bawah yang dapat dilihat pada gambar 2.1



Gambar 2.1 Empat Tipe Ukuran Kinerja

Kulit terluar menjelaskan kondisi keseluruhan dari sebuah bawang, jumlah sinar matahari, air, dan nutrisi yang telah diterima; bagaimana ia ditangani mulai dari proses panen hingga mencapai rak supermarket. Namun, semakin kita mengupas lapisan-lapisan bawang, kita akan menemukan lebih banyak informasi. Lapisan-lapisan tersebut mewakili indikator kinerja, dan yang menjadi intinya adalah *Key Performance Indicator*.

KRI memberi gambaran nyata apakah anda berjalan ke arah yang benar, namun, ukuran tersebut tidak memberitahukan apa yang anda perlukan untuk meningkatkan hasil. Jadi, KRI menyediakan informasi yang ideal bagi anggota dewan direksi (yaitu mereka yang tidak terlibat dalam manajemen sehari-hari). KRI umumnya mencakup periode yang lebih lama daripada KPI. Tidak seperti KPI yan ditinjau secara mingguan atau harian, KRI ditinjau menggunakan siklus bulanan atau tigabulanan. Ukuran bulanan, caturwulan, atau tahunan tidak dapat dijadikan KPI, karena tentu ukuran tersebut bukanlah kunci keberhasilan bisnis jika anda memonitornya setelah terjadi masalah. Dikatakan KPI jika proses monitoringnya langsung menindak lanjuti ketika terjadi suatu masalah. Beberapa perbedaan KRI dan KPI dapat dilihat pada Tabel 2.2.

Tabel 2.2 Perbedaan KRI dan KPI

KRI	KPI
Periode pengukuran bulanan, triwulan, caturwulan, 6 bulan, atau 1 tahun.	Periode pengukuran jangka pendek (mingguan, harian, atau *realtime*).
Sebagai informasi ringkas mengenai kemajuan utama (kritis) faktor keberhasilan organisasi.	Memberikan informasi yang lebih rinci mengenai kemajuan utama (kritis) faktor keberhasilan organisasi.
Umummnya, satu-satunya orang yang bertanggung jawab untuk KRI adalah CEO atau Kepala Sekolah dan Ketua yayasan yang tidak terlibat dalam manajemen sehari-hari	Tanggung jawab dibebankan kepada individu yang melakukan proses atau tim.
Tidak memberitahu apa yang diperlukan meningkatkan hasil.	Mengindikasikan secara jelas tindakan yang perlu dilakukan, memahami ukuran dan bagaimana memperbaiknya.
Fokus terhadap proses yang sudah dilakukan (masa lalu).	Fokus terhadap proses yang sedang dilakukan (masa kini) dan yang akan dilakukan (masa depan).
KRI tidak dapat membantu staf atau manajemen karena KRI tidak menjelaskan dimana yang harus diperbaiki	Semua staf dapat memahami ukuran tersebut dan tindakan perbaikan seperti apa yang perlu dilakukan
KRI merupakan hasil dari banyak aktivitas yang diatur melalui banyak ukuran kinerja	Mempengaruhi semua ukuran kinerja lainnya secara positif

Beberapa ukuran kinerja yang terletak diantara KRI dan KPI adalah indikator hasi (*Result Indicator / RI*) dan indikator kinerja (*Performance Indicator / PI*). *Performance Indicator* meskipun penting, bukanlah yang utama dalam bisnis. PI dapat membantu tim untuk menyelaraskan mereka dengan strategi organisasi. PI dan RI diukur harian, mingguan, dua mingguan, bulanan, atau terkadang caturwulanan. PI memusatkan pada aktivitas tertentu, sedangkan RI yang merupakan hasil lebih dari satu aktivitas. Analogi speedometer mobildapat menjelaskan perbedaan antara RI dan PI. Kecepatan sebuah mobil menempuh perjalanan adalah RI, karena kecepatan mobil adalah gabungan dari gigi yang digunakan oleh mobil tersebut dan beberapa banyak revolusi yang digunakan oleh mesin mobil setiap menitnya. PI adalah seberapa ekonomis mobil tersebut dikendarai (misalnya alat pengukur yang menunjukkan beberapa

kilometer jarak ditempuh untuk setiap liter bahan bakarnya). Atau seberapa panas mesin tersebut panas mesin tersebut pada saat mesin hidup (pengukur suhu). Beberapa perbedaan PI dan RI dapat dilihat pada Tabel 2.3

Tabel 2.3 Perbedaan PI dan RI

RI	PI
Tidak menjelaskan apa yang seharusnya dilakukan	Semua staf memahami tindakan apa yang perlu dilakukan untuk memperbaiki PI
Merupakan hasil lebih dari satu aktivitas	Memusatkan pada aktivitas tertentu

2.1.4 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan [1]. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DMBS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

2.1.4.1 DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [1]. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

2.1.4.2 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS [1]. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL yang secara umum terdiri dari 4 hal sebagai berikut:

1. Memasukkan data (*insert*)

```
INSERT INTO Tabel_mahasiswa
(nim, nama, tanggal_lahir)
VALUES
('10109240', 'Fahmi', '1991-05-03');
```

Query di atas digunakan untuk memasukkan data mahasiswa dengan NIM 10109240, nama Fahmi, dan tanggal lahir 3 Mei 1991 ke tabel "Tabel_mahasiswa".

2. Mengubah data (*update*)

```
UPDATE Tabel_mahasiswa
 SET
   Tanggal_lahir='1990-01-01'
  WHERE
    Nim='10109240';
```

Query di atas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM = 10109240 menjadi 1 Januari 1990 dalam tabel "Tabel_mahasiswa".

3. Menghapus data (*delete*)

```
DELETE FROM Tabel_mahasiswa
  WHERE
    Nim='10109240';
```

Query di atas dugunakan untuk menghapus data mahasiswa dengan NIM =10109240 dari tabel "Tabel_mahasiswa".

4. Menampilkan data (*select*)

```
SELECT nim,nama
 FROM
```

```
  Tabel_mahasiswa
WHERE
  Nim='10109240';
```

Query di atas dugunakan untuk menampilkan data mahasiswa yang terimpan dalam "Tabel_mahasiswa" dengan NIM = 10109240.

2.1.5 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [1]. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode 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 pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

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 (*reusable*).

2. Kecepatan pengembangan

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

3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang 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

   Meningkatkan kualitas perangkat lunak

2.1.5.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya

dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

   Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat *runtime* yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat *runtime* (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

   Metode atau operasi dapat berasal dari *event*, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya *Read, Write, Move, Copy,* dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

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

8. Antarmuka (*interface*)

   Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.1.6 Pengenalan UML

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada

saat itu, misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

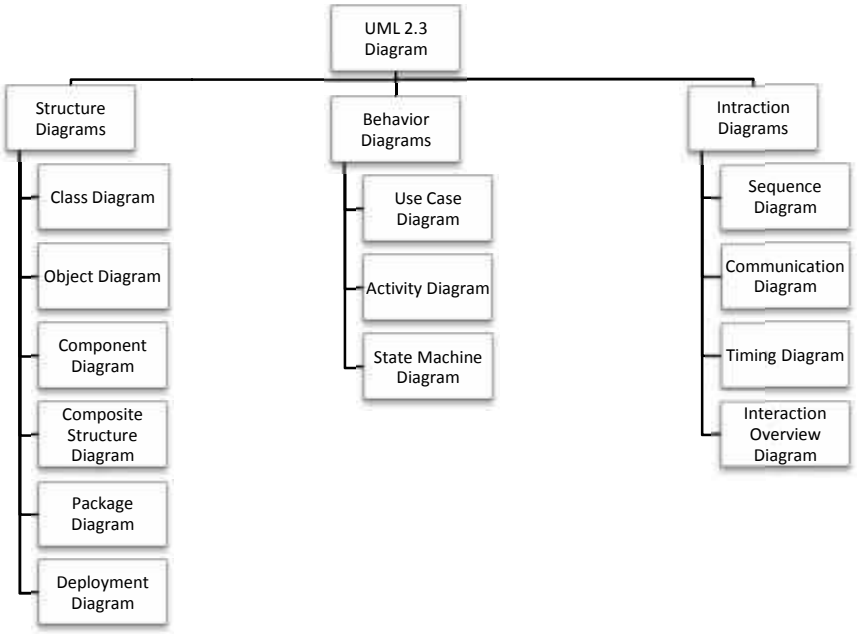
Pada perkembangan teknil pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan voisual untu menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan digaram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Seperti yang kita ketahui di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan, lingkunan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang perbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin digambarkan. Secara analogi jika dengan bahasa yang digunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalahhal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi di dalam bidang sistem informasi merupakan hal yang sangat wajar.

2.1.7 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.2.



Gambar 2.2 Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

2.1.7.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah

interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case.*

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.1.7.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

2.1.7.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.1.7.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas ebaiknya memiliki jenis-jenis kelas berikut :

1) Kelas Main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3) Kelas yang diambil dari pendefinisian *usecase*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4) Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan sat sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan coupling. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *choesion* yang kuat dan kadar *coupling* yang lemah.

Dalam class diagram terdapat beberapa relasi (hubungan antar class) yaitu;

1. Generalization dan Inheritence

   Diperlukan untuk memperlihatkan hubungan pewarisan (inheritance) antar unsur dalam diagram kelas. Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari kelas yang berada dalam hirarki pewarisannya.

2. Associations

   Hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui ekstensi class lain. Dalam notasi UML kita mengenal asosiasi 2 arah (bidirectional) dan 1 arah (undirectional).

3. Aggregation

Hubungan antar-class dimana class yang satu (part class) adalah bagian dari class lainnya (whole class).

4. Composition

   Aggregation dengan ikatan yang lebih kuat. Di dalam composite aggregation, siklus hidup part class sangat bergantung pada whole class sehingga bila objek instance dari whole class dihapus maka objek instance dari part calss juga akan terhapus.

5. Depedency

   Hubungan antar-class dimana sebuah class memiliki ketergantungan pada class lainnya tetapi tidak sebaliknya.

6. Realization

   Hubungan antar-class dimana sebuah class memiliki keharusan untuk mengikuti aturan yang ditetapkan class lainnya. biasanya realization digunakan untk menspesifikasikan hubungan antara sebuah interface dengan class yang mengimplementasikan interface tersebut.

2.1.8 Design Pattern

Design Pattern (Pola Desain) adalah suatu bagian dari teknologi pemroraman berorientasi objek. Pola desan adalah solusi yang sudah teruji untuk menyelesaikan permasalahan pemrograman. Pola desain merupakan sebuah pola atau cara untuk mendesai komponen-komponen dalam pemrograman berorientasi objek yang baik sehingga komponen-komponen yang ada dapat digunakan kembali untuk aplikasi yang lain. Pola desain dimulai dengan menelusuri solusi-solusi yang baik dari suatu masalah untuk menghasilkan pola yang baik.

Cara membuat sebuah pola desain komponen yang baik adalah dengan memperhatikan kemungkinan perubahan sistem yang mungkin terjadi di masa mendatang. Pola desain penting untuk dipelajari agar dapat merancang program berorientasi objek yang dapat digunakan kembali untuk aplikasi lain dan memudahkan kerja tim jika sudah menggunakan pola pembuatan program berorientasi objek yang sudah standar(mengurangi perbedaan cara berpikir antara

anggota tim). Pola yang baik memperhatikan *cohesion* (keeratan keterkaitan metode dalam sebuah kelas) dan *coupling* (keeratan hubungan antar kelas).

Design Pattern (Pola Desain) sebaiknya dipelajari ketika pemrograman berorientasi objek telah dipahami. Pola desain bertujuan agar komponen program bersifat *reusable* (dapat digunakan kembali berulang-ulang). *Framework* merupakan rancangan arsitektur perangkat lunak yang digunakan sebagai standar pembuatan program dan memudahkan pengguna dalam memelihara perangkat lunak yang dibuat dengan menggunakan *framework*. *Framework* yang berorientasi objek adalah salah satu bagian implementasi pola desain. Submodul-submodul berikutnya akan membahas pola-pola desain yang telah ada.

Pada sebuah pola biasanya memiliki atribut-atribut (hal-hal yang perlu diperhatikan) sebagai berikut:

1) Nama (*Name*)
   Nama yang diberikan pada pola
2) Tujuan (*Intent*)
   Merupakan pemaparan singkat mengenai permasalahan yang terjadi dan tujuan yang ingin dicapai
3) Alias (*Also Known As*)
   Nama alias yang diberikan pada pola, nama alias bersifat opsional
4) Motivasi (*Motivation*)
   Sebuah skenario yang menjelaskan permasalahan perancangan dan solusi yang diberikan pola desain
5) Kemampuan diaplikasikan (*Applicability*)
   Spesifikasi situasi yang sesuai untuk penerapan pola
6) Struktur (*Structure*)
   Gambar sruktur yang menjelaskan hubungan kelas dan objek dalam pola
7) Partisipasi (*Participants*)
   Berbagai kelas atau objek yang berpartisipasi dan berperan dalam pola
8) Kolaborasi (*Collaborations*)
   Bagaimana kolaborasi kerja sama dari partisipan untuk melaksanakan perannya masing-masing

9) Konsekuensi (*Consequences*)
   Bagaimana penapaian tujuan yang dilakukan pola dan konsekuensi penerapan pola
10) Implmentasi (*Implementation*)
   Berbagai hal yang berkaitan dengan penerapan pola
11) Contoh Kode Program (*Sample Code*)
   Contoh kode program yang merepresentasikan penerapan pola
12) Contoh Penggunaan Pola (*Known Uses*)
   Contoh-contoh penggunaan pola pada sistem yang riil
13) Pola-pola Lain yang Terkait (*Related Pattern*)
   Pola-pola lain yang berhubungan dengan pola ini

Pola desain dibagi menjadi beberapa kelompok sebagai berikut :

1. Pola yang Berbasis pada Proses Pembuatan (*Creational Patterns*)
   Pola disusun dengan menitikberatkan komponen-komponen (objek-objek) apa saja yang perlu untuk dibuat
2. Pola yang Berbasis pada Struktur (*Structural Patterns*)
   Pola disusun dengan menitikberatkan pada struktur spesifikasi komponen-komponen (kesesuaian objek dengan tindakan atau kemampuan yang diharapkan) pemrograman berorientasi objek
3. Pola yang Berbasis pada Kebiasaan (*Behavioral Patterns*)
   Pola disusun dengan menitikberatkan pada spesifikasi kelakuan sistem aplikasi yang akan dibuat apakah pola yang dibuat sudah sesuai dengan kelakuan atau kemampuan aplikasi yang diharapkan.
4. Pola yang berbasis pada Desain Arsitektur (*Architectural Design Pattern*)
   Pola disusun dengan menitikberatkan pada hirarti arsitektur (pemisahan blok-blok lingkut kerja).
   Model View Controller (MVC) merupakan salah satu contoh dari *Architectural Design Pattern.* Pada saat ini banyak *framework* yang didedikasikan pemrograman berorientasi objek.

Framework adalah kerangka kerja yang memudahkan *programmer* untuk membuat sebuah aplikasi sehingga *programmer* akan lebih mudah melakukan perubahan (customize) terhadap aplikasinya dan dapat memakainya kembali untuk aplikasi lain yang sejenis.

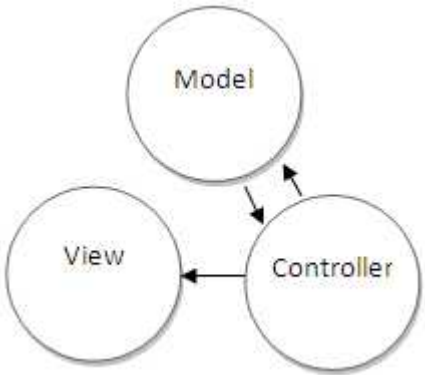
Contoh-contoh *framework* yang sering digunakan antara lain sebagai berikut

a) PHP : CodeIgniter, Smarty, cakePHP, Zend, Yii, dll
b) Java: Spring, Hibernate, Struts, dll
c) Ruby: Ruby on Rails, dll
d) Phyton: Django, dll
e) .NET: ASP.NET MVC Framework, dll

Konsep MVC bertujuan agar sebuah aplikasi dapat mudah dipelihara oleh orang-orang di dalam tim pengembangan yang berbeda spesifikasi pekerjaannya, misalnya *Database Administrator* (DBA) untuk mengurusi masalah basis data, blok *ontroller* untuk programmer, dan blok *view* untuk desainer antar muka (*interface designer*).

2.1.8.1 Model View Controller

Teknik pemrograman MVC (Model View Controller) merupakan teknik pemrograman yang populer saat ini, yang mengharapkan pemrogram secara disiplin untuk membagi program menjadi tiga bagian; model, view, dan controller, seperti gambar berikut :



Gambar 2.3 Arsitektur MVC

Model

Objek model adalah bagian dari aplikasi yang mengimplementasi logika untuk domain data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari database atau menyimpan data ke dalam database.

View

View adalah komponen yang menampilkan antarmuka untuk pengguna (user interface, UI) aplikasi. Antarmuka ini dibuat berdasarkan data dari model.

Controller

Controller merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan model, dan memilih view mana yang digunakan untuk merender data.

Dalam aplikasi MVC, view hanya menampilkan informasi, controller menangani dan menanggapi masukan dan interaksi dengan pengguna. Dengan pola MVC, kita membuat aplikasi dengan memisahkan berbagai aspek dari aplikasi (logika input, bisnis, dan antarmuka), ada bagian penampil data (View), penyedia atau pemroses data (model), dan pengendali program (controller).

Dengan pemisahan ini, maka pembuatan satu program dapat dibagi-bagi kepada beberapa orang. Setiap bagian atau komponen yang dibuat secara terpisah diharapkan dapat digunakan tidak hanya oleh satu program saja, tetapi oleh banyak program, terutama bagian model dan view. Teknik pemrograman MVC ini diimplementasi tidak hanya dalam PHP, tetapi suatu teknik pemrograman yang sudah dikenal dan digunakan hampir pada semua bahasa pemrograman yang ada.

2.2 Perangkat Lunak Pendukung

Adapun beberapa perangkat lunak pendukung dalam penyusunan Laporan Sistem Monitoring Perkembangan Santri Madrasah Mardliyah diantaranya:

2.2.1 PHP

PHP adalah suatu bahasa *server-side* yang di desain khusus untuk aplikasi web. PHP dapat disisipkan diantara HTML, dan karena bahasa server-side, maka

bahasa PHP akan dieksekusi di server, yang dikirimkan ke browser adalah “hasil jadi” bentuk HTML, dan kode PHP tidak akan terlihat.

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu terbaru. Semua script PHP dieksekusi pada server dimana sciprt tersebut dijalankan. Oleh karena itu, spesifikasi server lebih berpengaruh pada eksekusi dari script php daripada spesifikasi client. Namun tetap diperhatikan bahwa web yang dihasilkan tentunya harus dapat dibuka oleh browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client. PHP termasuk dalam *Open Source Product.* Jadi dapat menambah *source code* da dapat mendistribusikannya secara bebas.

2.2.2 MYSQL

Database MySQL merupakan sistem manajemen basisdata SQL yang sangat terkenal dan bersifat open source. MySQL dibangun, didistribusikan, dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang MySQL. MySQL mempunyai dua maca lisensi yaitu lisensi yang bersifat open source dengan menggunakan GNU (*General Public License*) dan lisensi kedua berupa *Standard Commercial License* yang dapat dibeli dari MySQL AB.

Dalam konteks bahasa SQL, informasi disimpan dalam tabel-tabel yang secara logis merupakan struktur dua dimensi yang tersimpan atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column).* Baris pada tabel disebut *insance* dari data sedangkan kolom sering disebut sebagai *atributes* atau *field.*

MySQL yang berjalan di sistem operasi Windows terbukti sangat stabil. MySQL versi Windows mempunyai fitur yang sama dengan MySQL versi Unix.

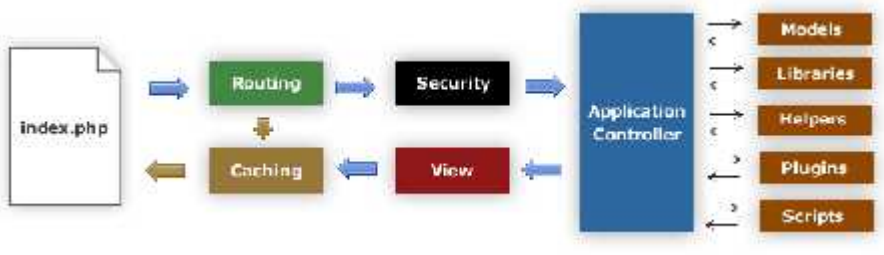
2.2.3 CodeIgniter

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu membuat program dari awal (*from scracth*), karena CI menyediakan sekumpulan librari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengkakses librarynya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan.

Framefork CI merupakan framework yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan mudah. Pendekatan dari CI sangatlah mudah, dari membuat sekadar tulisan sampai dengan yang kompleks dapat didekati dengan mudah. Tidak seperti framework yang lain, untuk mendapatkan tulisan Hello World di browser saja, kita harus menggunakan beberapa tahap. CI cukupd dengan satu file dan satu prosedur atau method.

2.2.3.1 Pemrograman MVC dengan CodeIgniter

CI adalah satu framework pemrograman PHP yang mengimplementasikan teknik pemrograman MVC. Berikut adalah arsitektur bagaimana CI mengimplementasikan MVC.



Gambar 2.4 Arsitektur CodeIgniter

Index.php berlaku sebagai controller di depan, yang menginisialisasi sumber daya yang dibutuhkan untuk menjalankan CodeIgniter

1) Router akan memeriksa request HTTP untuk menentukan apa yang harus dilakukan dengan request tersebut
2) Jika ada file cache, maka file cache langsung dikirimkan kepada browser, dengan memotong eksekusi sistem normal
3) Keamanan. Sebelum suatu aplikasi controller dipanggil, request HTTP dan data yang dikirimkan oleh pengguna disaring (filter) terlebih dahulu untuk keamanan
4) Controller akan memuat model, libari inti, helper, dan sumberdaya lainnya yang diperlukan untuk memroses request tersebut
5) View terakhir direnderkemudian dikirimkan kepada browser untuk dilihat. Jika caching didefinisikan, maka view akan dicache terlebih dahulu, setelah itu baru dikirimkan kepada browser.

Codeigniter (CI) bisa digunakan untuk belajar melakukan pemrograman dengan teknik pemrograman yang menggunakan konsep MVC (Model View Controller). Ci mengijinkan pemrogram untuk bekerja penuh di dalam controller, sambil memahami konsep MVC dengan benar.

2.3 Profil Instansi

Yayasan Super Insani adalah yayasan nirlaba yang mempunyai maksud dan tujuan di bidang sosial, kemanusiaan, dan keagamaan. Dalam menyelenggarakan kegiatannya di bidang sosial, kemanusiaan, dan keagamaan, yayasan memiliki divisi pendidikan dan divisi kesehatan. Divisi kesehatan memiliki dua sub divisi yaitu Klinik Habibah dan Rumah Sehat.

Divisi Pendidikan memiliki kegiatan; majelis taklim, perpustakaan, dan madrasah mardliyah. Madrasah Mardliyah yang beralamat di Jl. Sukaati Permai II No.11 Kota Bandung merupakan komunitas terbuka untuk umum yang bergerak pada bidang pendidikan. Program madrasah mardliyah meliputi; madrasah, pelatihan, perpustakaan, dan majelis taklim. Madrasah ini aktif sejak tahun 2008 dimulai dengan delapan orang santri; Fairuz, Azmi, Zaif, Yoga, Lina, Lala, Icha, dan Dhea dengan dua orang tenaga pengajari yaitu; Ustadz Eri dan Ustadz Kurnia dengan waktu belajar pada hari selasa, kamis dan sabtu. Kurikulum madrasah

mardliyah terdapat lima mata pelajaran yaitu; Al-Quran, Fiqih, Aqidah, Akhlaq, dan Life Skill. Madrasah ini berperan sebagai mitra orang tua dalam pendidikan agama, sebagai referensi orang tua dalam mengembangkan minat dan bakat anak dan sebagai madrasah keluarga.

Salah satu program Madrasah Mardliyah yaitu mengadakan kegiatan belajar mengajar pendidikan Agama Islam yang terdiri dari lima tingkat kelas; PAUD/TK, SD, SMP, SMA, dan tingkat ibu-ibu. Tujuan dari madrasah ini adalah menghasilkan santri yang cerdas, berakhlak, dan kreatif.

2.3.1 Visi dan Misi

Visi dan misi yang terdapat pada Madrasah Mardliyah adalah ;

1) Visi

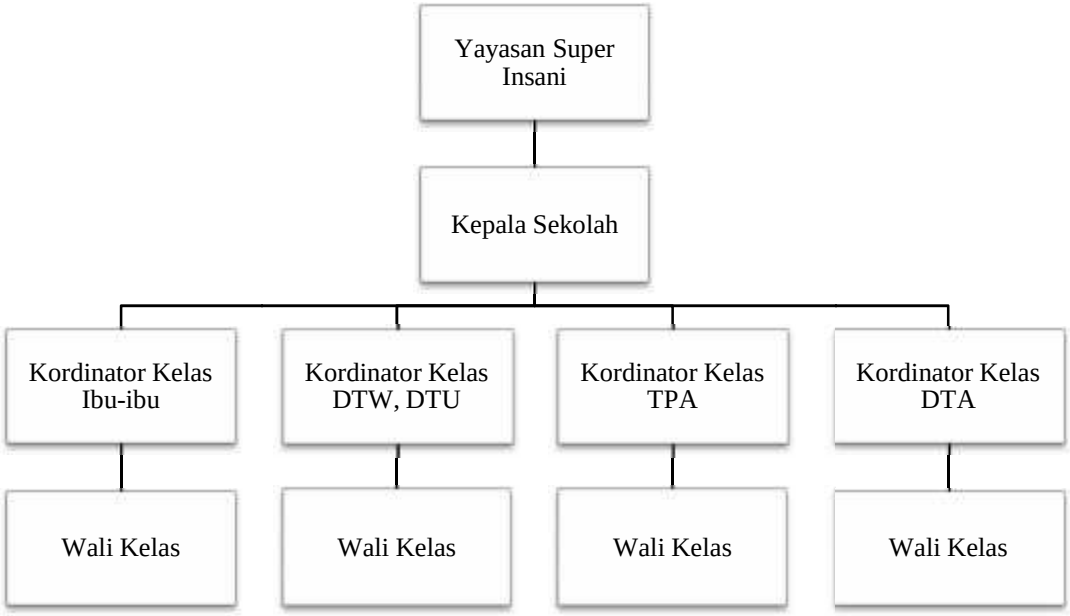
Membentuk dan membina keluarga serta masyarakat yang menerapkan nilai-nilai Islam dalam kehidupan sehari-hari.

2) Misi

1. Meningkatkan pemahaman masyarakat akan nilai-nilai islam melalui berbagai program pendidikan.
2. Meningkatkan kualitas sumber daya manusia berlandaskan kehidupan yang berawal dari keluarga yang islami
3. Mempererat silaturahmi sesama muslim yang memperkuat sendi kehidupan sosial
4. Meningkatkan wawasan keagamaan

2.3.2 Struktur Organisasi

Adapun struktur organisasi pada madrasah madrliyah dapat dilihat pada gambar II.1.



Gambar 2.5 Struktur Organisasi

2.3.3 Logo Instansi

Berikut logo madrasah mardliyah dapat dilihat pada Gambar 2.5



Gambar 2.6 Logo Babul Ilmi

Logo madrasah mardliyah adalah pintu dan buku. Pintu bermakna sebagai rumah yang terbuka bagi masyarakat. Dan buku bermakna sebagai ilmu. Jadi makna dari logo ini adalah rumah yang terbuka bagi masyarakat umum untuk mencari ilmu pendidikan keagamaan.