

BAB II

LANDASAN TEORI

2.1 Ayat Al-Qur'an Yang Berkaitan dengan Pelanggaran Kedisiplinan

Di dalam Al-Qur'an juga di jelaskan mengenai pelanggaran sebagai berikut :

وَإِذَا قِيلَ لَهُمْ لَا تُفْسِدُوا فِي الْأَرْضِ قَالُوا إِنَّمَا نَحْنُ مُصْلِحُونَ

“Dan bila dikatakan kepada mereka:"Janganlah kamu membuat kerusakan di muka bumi[24]". mereka menjawab: "Sesungguhnya Kami orang-orang yang Mengadakan perbaikan." (QS. Al baqarah[2]:11)

وَلَا تَلْبِسُوا الْحَقَّ بِالْبَاطِلِ وَتَكْتُمُوا الْحَقَّ وَأَنْتُمْ تَعْلَمُونَ

“Dan janganlah kamu campur adukkan yang hak dengan yang bathil dan janganlah kamu sembunyikan yang hak itu, sedang kamu mengetahui” (QS. Al baqarah[2]:42)

2.2 Teori yang Berhubungan dengan Sistem Secara Umum

2.2.1 Pengertian Sistem

Menurut Sutabri (2012: 22), bahwa sistem adalah suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut.

Sedangkan menurut Bambang Hartono (2013: 9), sistem adalah suatu himpunan dari berbagai bagian atau elemen, yang saling berhubungan secara terorganisasi berdasar fungsi-fungsinya, menjadi satu kesatuan.

2.2.2 Pengertian Informasi

Menurut Bambang Hartono (2013: 15), informasi adalah data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan keputusan saat ini atau di masa yang akan datang.

Sedangkan menurut Sutabri (2012: 29), bahwa informasi merupakan data yang telah diklasifikasikan atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

2.2.3 Pengertian Sistem Informasi

Menurut Sutabri (2012: 46), bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sedangkan menurut Bambang Hartono (2013: 16), sistem informasi adalah seperangkat komponen yang saling berhubungan, yang bekerja untuk mengumpulkan dan menyimpan data serta mengolahnya menjadi informasi untuk digunakan.

2.3 Teori yang Berhubungan dengan Teknik Analisa yang Digunakan

2.3.1 *Entity Relationship Diagram (ERD)*

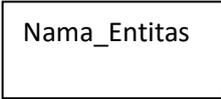
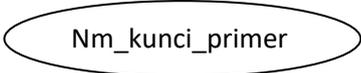
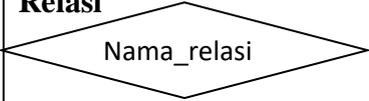
Menurut Al Fatta (2007:120-121), ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data. Akhirnya ERD bisa digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun. Aturan bisnis adalah batasan yang harus diikuti ketika sistem beroperasi.

Al-Bahra (2013:142), ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam kedalam sistem secara abstrak. ERD berbeda dengan DFD yang merupakan suatu jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan struktur-struktur relationship data. ERD memperlihatkan hubungan antara data store pada DFD. Hubungan ini tidak terlihat pada DFD, karena DFD hanya memusatkan perhatian pada fungsi-fungsi sistem bukan pada data yang dibutuhkan. Diagram hubungan entitas atau yang lebih dikenal dengan sebutan E-R Diagram, adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (storage data) dalam sistem secara

abstrak. Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data.

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relational, (Sukamto dan Shalahuddin, 2013:50-51).

Tabel 2.1 Simbol *Entity Relationship Diagram* (ERD)

Simbol	Keterangan
<p>Entitas/Entity</p> 	<p>Entitas</p> <p>Merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel</p>
<p>Atribut</p> 	<p>Field</p> <p>Kolom data yang butuh disimpan dalam suatu entitas.</p>
<p>Atribut Kunci Primer</p> 	<p>Field atau kolomdata</p> <p>Yang butuh disimpan dalam entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom.</p>
<p>Relasi</p> 	<p>Relasi</p> <p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</p>

Asosiasi/association <hr/>	Penghubung penghubung antara relasi dimana di ke dua ujungnya memiliki <i>multiplicacity</i> kemungkinan jumlah pemakai.
--------------------------------------	--

(Sumber: Sukamto dan Shalahuddin, *Rekayasa Perangkat Lunak*, 2013:..)

2.3.1.1 Kardinalitas (Cardinality)

Al Bahra (2013:147-148), Kardinalitas relasi menunjukkan jumlah maksimum tupel yang terdapat berelasi dengan entitas pada entitas lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat kardinalitas relasi, yaitu :

- 1) One to One : Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.
- 2) One to Many atau Many to One : tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.
- 3) Many to Many : tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua.

2.3.1.2 Tahapan Pembuatan ERD

Langkah-langkah teknis yang dilakukan untuk menghasilkan ERD adalah sebagai berikut :

- a. Melengkapi entitas dan relasi dengan atribut-atribut deskriptif (non-Key)
- b. Mengidentifikasi dan menetapkan seluruh entitas yang akan terlibat
- c. Menentukan atribut-atribut key (primary key) dari masing-masing entitas
- d. Mengidentifikasi dan menetapkan seluruh derajat/kardinalitas relasi antara entitas-entitas yang ada beserta foreign key-nya.
- e. Melengkapi entitas dan relasi dengan atribut-atribut deskriptif (non-key).

2.3.2 Data Flow Diagram (DFD)

Al-Bahra (2013:61), diagram aliran data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan. Sutabri (2004:163), dalam bukunya menjelaskan pendekatan analisis terstruktur diperkenalkan oleh DeMarco (1978) dan Gan Sarson (1979) melalui buku metodologi struktur analisis dan desain sistem informasi. Mereka menyarankan untuk menggunakan *data flow diagram* (DFD) dalam menggambarkan atau membuat model sistem.

Pengertian secara umum dari *data flow diagram* ini adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

Keuntungan dari DFD adalah memungkinkan untuk menggambarkan sistem dari level yang paling tinggi kemudian menguraikannya menjadi level yang lebih rendah (dekomposisi), sedangkan kekurangan dari DFD adalah tidak menunjukkan proses pengulangan (*looping*), proses keputusan dan proses perhitungan. Berikut menjelaskan level yang terdapat dalam DFD menurut Al-Bahra (2013:64-65), yaitu:

2.3.2.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem.

2.3.2.2 Diagram Nol/Zero (Overview Diagram)

Diagram Nol adalah diagram yang menggambarkan proses dari dataflow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan mengenai fungsi-fungsi utama atau proses yang ada, aliran data, dan eksternal entity.

2.3.2.3 Diagram Rinci (Level Diagram)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram zero atau diagram level di atasnya.

Tabel 2.2 Penomoran Level pada DFD

Nama Level	Nama Diagram	Nomor Process
0	Context	
1	Diagram 0	1.0, 2.0, 3.0, ...

2	Diagram 1.0	1.1, 2.1, 3.1, ...
2	Diagram 2.0	2.1, 2.2, 2.3, ...
2	Diagram 3.0	3.1, 3.2, 3.3, ...
3	Diagram 1.1	1.1.1, 1.1.2, ...
3	Diagram 2.1	1.2.1, 1.2.2, ...
3	Diagram 3.1	1.3.1, 1.3.2, ...
dst		

(Sumber: Al Bahra, *Analisis Dan Desain Sistem Informasi, ...*)

Level yang terdapat dalam satu level DFD seyogyanya tidak terdapat lebih dari 7 buah proses dan maksimal 9, bila lebih maka harus dilakukan dekomposisi.

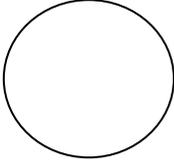
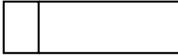
2.3.2.4 Blancing dalam DFD

Aliran data yang masuk ke dalam dan keluar dari suatu proses harus sama dengan aliran data yang masuk ke dalam dan keluar dari rincian proses pada level/tingkatan dibawahnya. Hal-hal yang perlu diperhatikan pada DFD yang memiliki lebih dari satu level sebagai berikut :

- a. Harus terdapat keseimbangan input dan output antara satu level dan level berikutnya.
- b. Keseimbangan antara level 0 dan level 1 terlihat dari input/output dari aliran data ke atau dari terminal pada level 0 sedangkan keseimbangan antara level 1 dan level 2 terlihat dari input dan output dari aliran data ke/ dari proses yang bersangkutan.
- c. Nama aliran data, data store dan terminal pada setiap level harus sama, apabila objeknya sama.

- d. Ada sumber buku yang menyatakan terminal tidak perlu digambarkan pada level 1,2 dan seterusnya namun untuk memperjelas diagram sebaiknya terminal tetap digambarkan pada level 1,2 dan seterusnya.

Table 2.3 Simbol *Data Flow Diagram* (DFD)

Simbol	Keterangan
	<p>External Entity</p> <p>Kesatuan luar merupakan kesatuan (<i>entity</i>) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.</p>
	<p>Data Flow</p> <p>Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem</p>
	<p>Process</p> <p>Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.</p>
	<p>Data Store</p> <p>Simpanan data merupakan simpanan data suatu <i>file</i>.</p>

(Sumber: Sukamto dan Shalahuddin, *Rekayasa Perangkat Lunak*,

2013:..)

2.4 Alat Bantu Aplikasi yang Digunakan dalam Pembuatan Sistem

2.4.1 Java

Java adalah bahasa pemrograman yang dapat di jalankan di berbagai jenis komputer dan berbagai sistem informasi termasuk telepon genggam. *Java* dikembangkan oleh Sun Microsystems dan dirilis tahun 1995. *Java* merupakan suatu teknologi perangkat lunak yang digolongkan multi platform. Selain itu, *Java* juga merupakan suatu platform yang memiliki virtual machine dan library yang di perlukan untuk menulis dan menjalankan suatu program. (Wahana Komputer, 2010 : 1)

2.4.2 Pengertian Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Tidak hanya menjadi sistem operasi di smartphone, saat ini *android* menjadi pesaing utama dari *apple* pada sistem operasi Tablet PC. Pesatnya pertumbuhan *android* karena merupakan *platform* yang sangat lengkap baik itu sistem operasinya, aplikasi dan tool pengembangan, market aplikasi *android* serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga *android* terus berkembang baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia. (Nazruddin Safaat H, 2014 : 1).

2.4.3 Android SDK (Software Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa *java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *realse* oleh *google*. Saat ini disediakan *Android SDK* sebagai alat bantu dan API untuk memulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*. Sebagai *platform* aplikasi-netral, *android* memberikan

kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *handphone*. Berikut beberapa fitur *android* yang paling penting adalah *Framework* aplikasi yang mendukung penggantian komponen dan reusable, Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*, *Integrataed browser* berdasarkan *engine open source WebKit*, Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1,0* (Opsional akselerasi *hardware*), *SQLite* untuk penyimpanan data, Media support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung *hardware*), *Bluetooth*, *Edge*, *3G*, dan *Wifi* (tergantung *hardware*), Kamera, Gps, Kompas, dan *accelerometer* (tergantung *hardware*), Lingkungan *development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk *IDE Eclipse*. (Nazruddin Safaat H, 2014 : 5).

2.4.4 ADT (*Android Development Tools*)

ADT (*Android Development Tools*) adalah *plugin* yang didesain untuk *IDE Eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi *android* dengan menggunakan *IDE Eclipse*. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan dalam membuat aplikasi projek *android*, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga dapat melakukan *running* aplikasi menggunakan *Android SDK* melalui *Eclipse* dengan ADT juga dapat melakukan pembuatan *package android* (.apk) yang digunakan untuk distribusi aplikasi anroid yang dirancang.

Mengembangkan aplikasi *android* dengan menggunakan ADT di *eclipse* sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi *android*. Berikut adalah versi ADT untuk *eclipse* yang sudah rilis : ADT 12.0.0 (Juli 2011), ADT 11.0.0 (Juni 2011), ADT 10.0.1 (Maret 2011), ADT 10.0.0 (Februari 2010), ADT 9.0.0 (Januari 2011), ADT 8.0.1 (Desember 2010), ADT 8.0.0 (Desember 2010), ADT 0.9.9 (September 2010), ADT 0.9.8 (September

2010), ADT 0.9.7 (Mei 2010), ADT 0.9.6 (Maret 2010), ADT 0.9.5 (Desember 2009), ADT 0.9.4 (Oktober 2009). (Nazruddin Safaat H, 2014 : 6).

2.4.5 Arsitektur Android

Secara garis besar arsitektur android dalam buku (Nazruddin Safaat H, 2014 : 6) Dapat dijelaskan dan digambarkan sebagai berikut :

2.4.5.1 Application dan Widgets

Application dan *Widgets* ini adalah *layer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien, email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java*.

2.4.5.2 Application Frameworks

Applications Frameworks adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk didalam *application frameworks* adalah sebagai berikut :

- a) *Views*
- b) *Content Providers*
- c) *Resource Manager*
- d) *Notification Manager*
- e) *Activity Manager*

2.4.5.3 Libraries

Libraries adalah *layer* dimana fitur-fitur *android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan

diatas *kernel*, Layer ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta :

- a) *Libraries* media untuk pemutaran media audio dan video
- b) *Libraries* untuk manajemen tampilan
- c) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
- d) *Libraries SQLite* untuk dukungan database
- e) *Libraries* SSL dan *Webkit* terintegrasi dengan *web browser* dan security
- f) *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view*
- g) *Libraries* 3D yang mencakup implementasi Open GL ES 1.0 API'sz

2.4.5.4 Android Run Time

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosenya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Didalam android run time dibagi dua bagian yaitu :

- a) *Core Libraries* : aplikasi *android* dibangun dalam bahasa *java*, sementara *dalvik* sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa *java/c* yang ditangani oleh *Core Libraries*.
- b) *Dalvik Virtual Machine* : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan threading dan manajemen tingkat rendah.

2.4.5.5 Linux Kernel

Linux kernel adalah layer dimana inti dari sistem operasi android itu berada. Berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi *android* lainnya. *Linux kernel* yang digunakan adalah *linux kernel realease 2.6*.

2.4.5.6 Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya dipackage oleh tools yang dinamakan “*apt tools*” ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang disebut dengan aplikasi, dan nantinya dapat di install di perangkat *mobile*. (Nazruddin Safaat H, 2014 : 9).

Ada empat jenis komponen pada aplikasi android yaitu :

a) *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bias jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan *desain* dari aplikasi tersebut.

b) *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya.

c) *Broadcast Receiver*

Broadcast receiver berfungsi menerima dan berinteraksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu berubah, *baterai low*, gambar telah selesai diambil oleh kamera, atau perubahan referensi bahasa yang digunakan.

d) *Content Provider*

Content Provider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta

(Map), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari content provider.

2.5 *PHP*

Menurut Agus Saputra (14-15: 2013), PHP merupakan singkatan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP berbeda kondisi dengan HTML karena HTML digunakan sebagai pembangunan atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya, sehingga dengan adanya PHP sebuah *web* akan mudah di-*maintenance*. PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *Server Side Scripting* yang artinya dalam setiap menjalankan PHP wajib membutuhkan *web server* dalam menjalankannya.

2.6 *MySQL*

Agus Saputra (2013:31), menjelaskan bahwa *MySQL* bekerja menggunakan *SQL Language (Structure Query Language)* pada umumnya perintah yang paling sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus), selain itu *SQL* menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data. Rosa (2014 : 46). menambahkan *SQL (Structure Query Language)* adalah bahasa yang digunakan untuk mengelola data pada *RDBMS*, *SQL* awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

2.7 Web Service

Menurut Jurnal Generic, Hartati Deviana (ISSN: 1907-4093). *Web service* merupakan suatu komponen software yang merupakan *selfcontaining*, aplikasi modular *self-describing* yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada web. *Web service* adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan *transactional web*, yaitu kemampuan web untuk saling berkomunikasi dengan pola *program-to-program* (P2P).

Dan sebuah buku yang di terbitkan oleh O'Reilly, penulis David A Chappel dan Tyler Jewell mengemukakan bahwa web service adalah kumpulan logika bisnis dalam internet yang dapat di akses melalui protokol internet. Dalam buku tersebut juga diuraikan Terdapat tiga teknologi dalam Web service yaitu, Simple Object Acces Protocol (SOAP), Web Service Description Language (WSDL), dan Universal Description, Discoveri, Integration (UDDI) (ISSN 2301-8402)

2.8 Eclipse

Eclipse adalah *tools* yang berisi *IDE eclipse* yang digunakan untuk membuat coding *android*. (Nazruddin Safaat H, 2014 : 15).

2.9 Pengertian Basis Data

Menurut buku (Rosa dan Shalahuddin, 2014 : 43) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat di butuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat di akses dengan mudah dan cepat.

2.10 DBMS

DBMS (Database Management System) atau dalam bahasa indonesia sering di sebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi

yang digunakan untuk menyimpan, mengelola, dan menampilkan data. (Rosa dan Shalahuddin, 2014: 44)

2.9 Tinjauan Pustaka

Berikut adalah tinjauan pustaka atau penelitian terdahulu tentang sistem pelanggaran kedisiplinan siswa :

Tabel 2.4 Tinjauan Pustaka

No	Nama Peneliti	Judul	Metode	Masalah	Hasil
1	Fitri Atmasari, Bambang Eka Purnama dan Indah (2013)	Sistem Informasi <i>Point</i> Pelanggaran berbasis SMS <i>gateway</i> pada MTs Ma'arif 2 Pacitan	<i>Waterfall</i>	Proses penyampaian informasi pelanggaran kepada orang tua masih dalam bentuk surat edaran.	Sistem Informasi <i>point</i> pelanggaran berbasis sms <i>gateway</i> pada MTs Ma'arif 2 Pacitan
2	Deni Multazam, Agus Prasetyo Utomo (2014)	Sistem Informasi <i>Monitoring</i> Studi Siswa Sma Berbasis Kurikulum Nasional 2013	<i>Prototype</i>	Proses kalkulasi dan informasi pelanggaran lambat	Aplikasi sistem informasi <i>monitoring</i> studi siswa berbasis kurikulum nasional 2013 pada MA Darul Ulum
3	Sarifudin Siswanto (2015)	Sistem Informasi Poin Pelanggaran dan Prestasi Siswa Berbasis Sms <i>Gateway</i> Pada SMK 3 Pacitan	<i>Waterfall</i>	kurangnya hubungan komunikasi antara siswa, guru dan orang tua dalam pengawasan siswa di sekolah	Sistem Informasi Poin Pelanggaran dan Prestasi Siswa Berbasis Sms <i>Gateway</i> Pada SMK 3 Pacitan
4	Dafit Nur Hidayanto, Tri Irianto Tj, Berliana K	Sistem Informasi Tata Tertib Siswa pada Smp Negeri 1 Jepara Dengan Menggunakan Sms <i>Gateway</i>	<i>Waterfall</i>	Informasi pelanggaran kedisiplinan masih dalam bentuk surat edaran	Sistem Informasi Tata Tertib Siswa pada Smp Negeri 1 Jepara dengan Meng

	Riasti (2013)				gunakan Sms Gateway
5	Patmi Kasih, Yuni Lestari (2013)	Aplikasi Penghitung <i>Point</i> Pelanggaran Siswa Sebagai Sistem Pendukung Keputusan Bagi Badan Konseling Sekolah	<i>Prototype</i>	Komunikasi antara Komponen Sekolah dan Orang tua tidak terjalin dengan baik	Aplikasi Penghitung <i>Point</i> Pelanggaran Siswa Sebagai Sistem Pendukung Keputusan pada SMK N 1 Tanah Grogot-Kaltim

