

BAB II

LANDASAN TEORI

2.1 Konsep Sistem Informasi

Sistem Informasi merupakan gabungan dari kata Sistem dan Informasi, Sistem adalah kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan. Selain itu sistem juga dapat didefinisikan sebagai sekumpulan objek-objek yang saling berelasi dan berinteraksi, serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan yang telah ditetapkan. Sedangkan Informasi adalah data yang sudah diproses menjadi bentuk yang berguna bagi pemakai, dan mempunyai nilai pikir yang nyata bagi pembuatan keputusan pada saat sedang berjalan atau untuk masa depan (Loveri 2016). Jadi Sistem Informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan. Sistem Informasi yaitu sistem di dalam suatu organisasi yang merupakan rangkaian dari orang-orang, data, proses, *interface*, teknologi, prosedur-prosedur yang mengeluarkan informasi untuk mencapai tujuan dalam pemecahan masalah dan/atau pengambilan keputusan. Sistem Informasi juga merupakan sistem di dalam 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 (Ariska et al. 2016).

Selain itu menurut beberapa ahli definisi sistem informasi sebagai berikut:

Menurut Robert yang dikutip oleh Minarni dan Saputra (2011), Sistem informasi adalah suatu sistem di dalam 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.

Menurut Leitch Davis yang dikutip oleh Minarni dan Saputra (2011), Sistem informasi adalah “suatu sistem di dalam sebuah organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian yang mendukung operasi dan bersifat manajerial dan kegiatan strategis yang diperlukan bagi pihak luar tertentu” (Minarni dkk 2011).

2.2 Komponen Sistem Informasi

Sistem informasi mengandung komponen-komponen seperti berikut:

1. Perangkat keras (*hardware*), yang mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yakni semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Basis Data (*database*), yaitu kumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*sources*) dipakai secara bersama atau diakses oleh sejumlah pemakai (Loveri 2016).

2.3 Perancangan Sistem Informasi

2.3.1 Analisa dan Perancangan

Analisa dan perancangan berkaitan dengan bagaimana suatu sistem akan diwujudkan dengan menggunakan teknologi berbasis komputer, beberapa *tools* yang digunakan untuk perancangan sistem adalah sebagai berikut:

2.3.1.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) disebut juga dengan Diagram Arus Data (DAD). DFD adalah: suatu model logika data atau proses yang dibuat

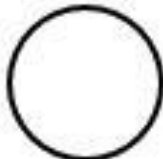
untuk menggambarkan: darimana asal data, dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

DFD yang di dalam bahasa Indonesia disebut sebagai DAD (Diagram Arus Data) memperlihatkan gambaran tentang masukan- proses-keluaran dari suatu sistem/perangkat lunak, yaitu obyek-obyek data mengalir ke dalam perangkat lunak, kemudian ditransformasi oleh elemen-elemen pemrosesan , dan obyek-obyek data hasilnya akan mengalir keluar dari sistem/perangkat lunak. Obyek-obyek data dalam penggambaran DFD biasanya direpresentasikan menggunakan tanda panah berlabel, dan transformasi-transformasi biasanya direpresentasikan menggunakan lingkaran-lingkaran yang sering disebut sebagai gelembung. DFD pada dasarnya digambarkan dalam bentuk hirarki, yang pertama sering disebut sebagai DFD level 0 yang menggambarkan sistem secara keseluruhan sedangkan DFD-DFD berikutnya merupakan penghalusan dari DFD sebelumnya.

DFD menggunakan empat buah simbol, yaitu: semua simbol yang digunakan pada CD ditambah satu simbol lagi untuk melambangkan data store. Ada dua teknik dasar penggambaran simbol DFD yang umum dipakai: pertama adalah Gane and Sarson sedangkan yang kedua adalah Yourdon and De Marco. Perbedaan yang mendasar pada teknik tersebut adalah lambang dari simbol yang digunakan.

Yourdon and De Marco menggunakan lambang lingkaran untuk menggambarkan *process* dan menggunakan lambang garis sejajar untuk menggambarkan *data store*. Sedangkan untuk simbol *external entity* dan simbol *data flow* kedua teknik tersebut menggunakan lambang yang sama yaitu: segi empat untuk melambangkan *external entity* dan anak panah untuk melambangkan *data flow* (Soufitri 2019).

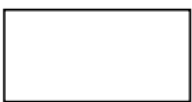
Tabel 2. 1 *Data Flow Diagram*

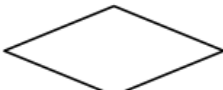
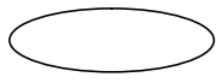
Gambar	Nama	Keterangan
	Terminator	Kesatuan diluar sistem (<i>external entity</i>) yang memberikan input ke sistem / menerima output dari sistem berupa oraganisasi, orang, atau sistem lain.
	Proses	Aktivitas yang mengelolah input menjadi output.
	Data Flow	Aliran data pada sistem (antar proses, antara proses dan terminator, serta antara proses dan data store).
	Data Store	Penyimpanan data pada database, biasanya berupa tabel.

2.3.1.2 Entity Relationship Diagram (ERD)

Model ERD berisi komponen-komponen entitas dan himpunan relasi yang masingmasing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara *entity-entity* yang ada dengan atribut-atributnya. Selain itu juga bisa menggambarkan hubungan yang ada dalam pengolahan data, seperti hubungan *many to many*, *one to many*, *one to one* (Sukrianto 2017).

Tabel 2. 2 Entity Relationship Diagram

Gambar	Keterangan
	Entitas , adalah suatu objek unik yang dapat diidentifikasi dalam lingkungan pemakai.

	Relasi , menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
	Atribut , adalah properti dari entitas atau tipe relasi.
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dengan entitas dan atribut.

2.3.2 Metode Pengembangan Sistem

Metode pengembangan sistem informasi merupakan suatu metode yang digunakan untuk melakukan pengembangan sistem informasi berbasis komputer. Beberapa metode yang digunakan dalam pengembangan sistem adalah sebagai berikut:

2.3.2.1 SDLC (System Development Life Cycle)

Metode System Development Life Cycle (SDLC) merupakan pengembangan berfungsi sebagai sebuah mekanisme untuk mengidentifikasi perangkat lunak. Proses pengembangan sistem melewati beberapa tahapan mulai dari sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara (Rudi, Arief, and Victor 2016).

Siklus hidup pengembangan sistem adalah suatu metode yang biasa digunakan dalam pengembangan sistem, yang terdiri dari 7 langkah utama yaitu:

1. Tahap kebijakan dan perencanaan sistem

Merupakan tahap awal dalam siklus pengembangan sistem, sebelum suatu sistem informasi dikembangkan, umumnya terlebih dahulu dimulai dengan adanya suatu analisis, kebijakan dan perencanaan untuk mengembangkan sistem itu. Tanpa adanya perencanaan sistem yang baik, pengembangan sistem tidak akan dapat berjalan sesuai dengan yang diharapkan. Perencanaan sistem (*System Planning*) merupakan pedoman untuk melakukan pengembangan sistem.

2. Tahap analisis sistem

Dilakukan setelah tahap perencanaan sistem dan sebelum tahap disain sistem. Pada tahap ini diberikan penilaian terhadap sistem yang lama dan dipelajari kelemahan yang ada serta mencari beberapa alternatif pemecahan masalah. Di dalam tahap ini terdapat langkahlangkah dasar yang harus dilakukan yaitu:

- a) Identify, yaitu mengidentifikasi masalah.
- b) Understand, yaitu memahami kerja dari sistem yang ada.
- c) Analyse, yaitu menganalisis sistem.
- d) Report, yaitu membuat laporan hasil analisis.

3. Tahap perancangan sistem secara umum

Setelah tahap analisis dilakukan, maka dilakukan pengembangan sistem secara umum dan menjelaskan informasi yang dihasilkan sistem tersebut.

4. Tahap seleksi sistem

Tahap ini mencari beberapa penyebab permasalahan pada sistem lama dan memilih satu pemecahan masalah dari beberapa alternatif yang ada.

5. Tahap perancangan sistem secara terinci

Tahap ini merupakan perincian dari sistem secara umum, disain sistem secara umum ditransformasikan ke dalam bentuk yang lebih spesifik untuk membangun sebuah sistem.

6. Tahap implementasi sistem

Tahap ini merupakan tahap dimana suatu sistem siap untuk dioperasikan.

Tahap ini terdiri dari langkah - langkah sebagai berikut:

- a) Menerapkan rencana implementasi
- b) Melakukan kegiatan implementasi
- c) Tindak lanjut implementasi.

7. Tahap perawatan sistem

Tahap ini adalah merupakan tahap akhir dalam sebuah pengembangan sistem. Dimana pada tahap ini hanya lebih ditekankan kepada pemeliharaan sistem, perawatan database dan menjaga kerja sistem yang sudah direncanakan di tahap pertama (Loveri 2016).

2.3.2.2 Model Waterfall

Model waterfall sering disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*) (A.S, R., & Shalahuddin 2018).

Metode waterfall merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Metode Waterfall memiliki tahapan-tahapan sebagai berikut (Sasmito 2017):

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

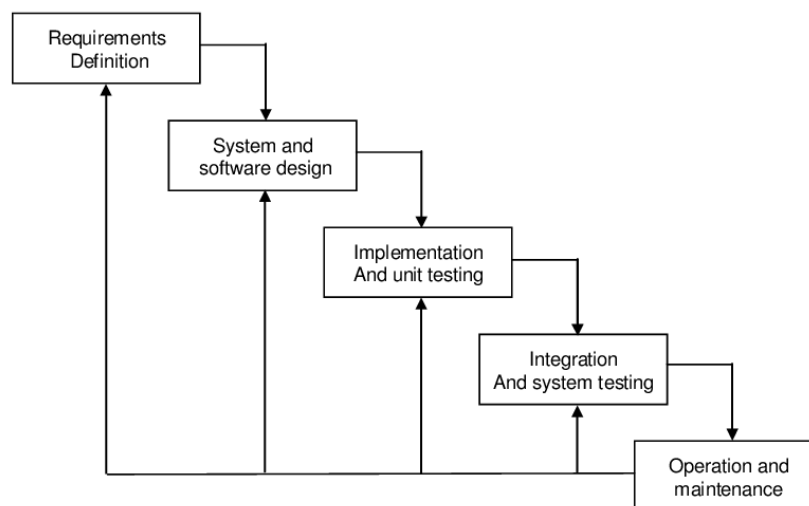
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak.

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem dan meningkatkan layanan sistem sebagai kebutuhan baru.



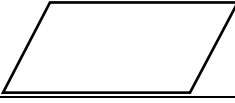
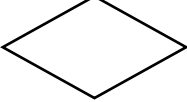
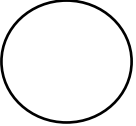
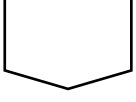
Gambar 2 1 Metode *Waterfall*

2.3.2.3 Model *Flowchart*

Flowchart adalah suatu teknik untuk menulis algoritma menggunakan simbol *chart* (gambar) dan garis panah ($\rightarrow$). Simbol *chart* mewakili fungsi langkah pengerjaan, sedangkan simbol garis panah mewakili alir pengerjaan simbol *chart*. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif alternatif lain dalam pengoperasian. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program (Wisnu Yudo Untoro 2010).

Tabel 2. 3 Tabel *Flowchart*

Gambar	Nama	Keterangan
	<i>Terminator</i>	Permulaan/akhir program
	Garis Alir (<i>Flowchart</i>)	Arah aliran program

	<i>Process</i>	Proses perhitungan/proses pengolahan data
	<i>Input/Output data</i>	Proses <i>input/output</i> data, parameter, informasi
	<i>redefined process (Sub Program)</i>	Permulaan sub program/proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>On Page Connector</i>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<i>Off page Connector</i>	Penghubung bagian-bagian flowchart yang berada pada satu halaman yang berbeda

2.4 Teknologi Informasi

Teknologi informasi merupakan teknologi yang menunjang sistem informasi berbasis komputer

2.4.1 Website

Website adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) didalamnya yang menggunakan protokol HTTP (hyper text transfer protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*. Beberapa jenis *browser* yang populer saat ini di antaranya : Internet Explorer yang diproduksi oleh Microsoft, Mozilla Firefox, Opera dan Safari yang diproduksi oleh Apple. *Browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat didalam aplikasi

browser yang biasa disebut *web engine*. Semua dokumen *web* ditampilkan dengan cara diterjemahkan (Suhartanto 2012).

2.4.1.1 CodeIgniter

CodeIgniter merupakan sebuah *framework* yang dibuat dengan menggunakan bahasa PHP, yang dapat digunakan untuk pengembangan *web* secara cepat. Adapun *framework* sendiri dapat diartikan sebagai suatu struktur pustaka-pustaka, kelas-kelas dan infrastruktur *run-time* yang dapat digunakan oleh programmer untuk mengembangkan aplikasi *web* secara cepat. Tujuan penggunaan *framework* adalah untuk mempermudah pengembang *web* mengembangkan aplikasi *web* yang *robust* secara cepat tanpa kehilangan fleksibilitas.

Pola desain dalam pengembangan *web* dengan *CodeIgniter* menggunakan MVC (*Models-View-Controller*). Dimana aplikasi yang dibuat akan dipisahkan antara logika bisnis dan presentasinya, sehingga memungkinkan *web programmer* dan *web designer* bekerja secara terpisah antara satu dengan yang lain. Agar bisa mengembangkan *web* dengan *CodeIgniter*, maka perlu dipahami terlebih dahulu konsep MVC dan struktur direktori dari *CodeIgniter*.

1. Models

Models merupakan *class* PHP yang didesain untuk bekerja dengan informasi yang ada didalam *basidata* seperti operasi menambah, mengubah, menghapus dan menampilkan informasi yang ada didalam *basidata*.

2. Views

Bagian ini bertanggung jawab terhadap tampilan informasi yang diperoleh dari *basidata/models*. Pada konsep MVC, *views* merupakan bagian presentasi yang berisi code HTML. Ketika ingin mengubah tampilan dari aplikasi yang dibuat, cukup mengubah bagian *views*.

3. Controller

Controller bisa dikatakan sebagai jantung dari aplikasi *web* yang dibuat, *controller* berisi logika bisnis yang menginstruksikan model untuk mengambil informasi dari basisdata dan kemudian ditampilkan

4. Fitur-fitur CodeIgniter

Berikut adalah fitur-fitur yang dimiliki oleh CodeIgniter:

- a) *Model-View-Controller*, merupakan fitur yang digunakan untuk mengembangkan aplikasi web dengan memisahkan antara logika bisnis dan presentasi.
- b) Support terhadap PHP 4 dan PHP 5
- c) *Template Engine Class*, merupakan fitur CodeIgniter untuk memisahkan data dan presentasi.

Scaffolding, merupakan fitur yang ada di *framework* CodeIgniter yang memungkinkan untuk membangun aplikasi cepat dan mudah untuk menambah, mengubah atau menghapus informasi dari basisdata (Afuan Lasmedi 2010).

2.4.1.2 PHP (PHP Hypertext Processor)

PHP singkatan dari *PHP Hypertext Processor* yang digunakan sebagai bahasa script server-side dalam pengembangan *Web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *Web* dapat dibuat dinamis sehingga *maintenance* situs *Web* tersebut menjadi lebih mudah dan efisien. PHP merupakan *software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya <http://www.php.net>. PHP ditulis dengan menggunakan bahasa C.

Sintaks Program/Script ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok script PHP:

1. `<?php.....?>`
2. `<script
language="PHP">.....</script>`
3. `<?.....?>` (Suhartanto 2012).

2.4.2. QR Code

QR Code adalah suatu jenis image dua dimensi yang menampilkan data berupa teks. Dengan *QR Code* pengguna dapat mengakses suatu web lebih cepat daripada harus melakukan secara manual. *QR Code* dapat didaur ulang dan digunakan kembali (Banjar Jawi 2018).

QR Code harus pada tempat yang terang atau pencahayaan yang cukup agar kode pada *QR Code* terbaca oleh pemindai. *QR Code* mampu menampung banyak informasi dengan 7.089 karakter numerik dan 4296 karakter alfanumerik. Pemindaian *QR Code* juga dapat digunakan sebagai sistem keamanan untuk tempat tertentu yang boleh dimasuki orang-orang tertentu seperti ruang riset ataupun ruang pusat (Banjar Jawi 2018).

QR Code adalah simbol dua dimensi yang dikembangkan oleh Denso Wave 1994 dengan tujuan utama sebagai simbol yang dapat dengan mudah diinterpretasikan oleh alat *scanner*. QR adalah merek dagang terdaftar dari perusahaan Jepang Denso Wave, anak perusahaan dari Toyota, yang menemukan teknologi tersebut pada tahun 1994 untuk melacak bagian dalam perakitan kendaraan. Denso Wave memilih untuk tidak melaksanakan paten atas teknologi ini dan mempromosikan penggunaannya secara luas. Di Jepang, *QR Code* telah digunakan secara luas dalam kegiatan pemasaran sejak awal 1990 karena kemampuannya untuk menciptakan interaksi langsung dengan konsumen.

QR Code mempunyai karakteristik yang berbeda dengan barcode tradisional, antara lain:

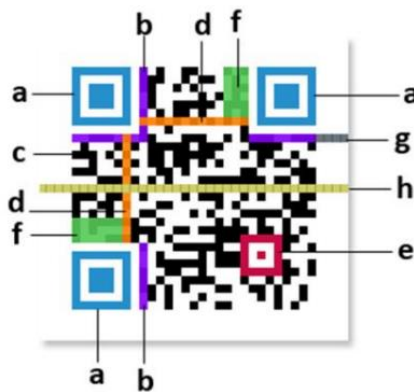
1. Mampu menyimpan data tersandi dalam kapasitas besar *QR Code* mampu menyandikan berbagai macam tipe data seperti numeris, karakter, Kanji, Hiragana, simbol, biner, bahkan mampu menyandikan 7089 karakter hanya dalam satu simbol. Berbeda dengan barcode biasa yang hanya mampu menyimpan informasi sebesar 20 digit.
2. Ukuran *printout* yang kecil *QR Code* mampu menyandikan data hanya dengan membutuhkan sepersepuluh ruangan yang dibutuhkan oleh barcode biasa.

3. Mampu menyandikan Kanji dan Kana
4. Tahan terhadap kotoran dan kerusakan *QR Code* mempunyai koreksi *error*, dimana data dapat direstore walaupun sebagian simbol kotor ataupun rusak.
5. Mampu terbaca pada arah manapun (360 derajat) *QR Code* mampu dibaca dalam berbagai arah (omni direksional) secara cepat. *QR Code* mempunyai pola untuk mendeteksi posisi pada tiga pojok simbol.
6. Kepadatan yang tinggi (rata-rata 100 kali lebih tinggi daripada barcode linear).
7. Pembacaan berkecepatan tinggi.
8. Memiliki keunggulan dalam unjuk kerja dan aspek fungsional (Ariska et al. 2016).



Gambar 2 2 Gambar *QR Code*

QR Code sendiri memiliki struktur yang mempunyai fungsi masing-masing, berikut merupakan struktur-struktur dari *QR Code*:



Gambar 2 3 Struktur *QR Code*

Berikut ini merupakan penjelasan yang berkenaan dengan gambar struktur *QR Code* di atas:

- a) Finder Pattern merupakan pola untuk mendeteksi posisi dari *QR Code*.
- b) Format Information berfungsi untuk informasi tentang error correction level dan mask pattern.
- c) Data merupakan daerah tempat data tersimpan atau data dikodekan.
- d) Timing Pattern merupakan pola yang digunakan untuk identifikasi koordinat pusat dari *QR Code*, dibuat dalam bentuk modul hitam putih bergantian.
- e) Alignment Pattern merupakan pola yang digunakan untuk memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
- f) Version Information merupakan versi dari sebuah *QR Code*, versi terkecil adalah 1 (21×21) modul dan versi terbesar adalah 40 (177×177) modul.⁵⁷
- g) Quiet Zone merupakan daerah kosong dibagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor CCD.
- h) *QR Code* version merupakan versi *QR Code*. Pada contoh gambar, versi yang digunakan adalah versi 3 (29×29 modul) (Putri 2020)

Salah satu contoh penggunaan dari *QR Code* yang sering kita jumpai adalah sebagai salah satu alternatif metode pembayaran secara digital yang

dimana pengguna hanya perlu memindai kode dan saldo digital akan berkurang sesuai dengan harga pembelian.

2.5 Database

Basis data adalah kumpulan file-file yang saling berhubungan atau relasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Suatu sistem manajemen basis data berisi satu kumpulan data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem manajemen basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data (Fathansyah, 2009).

Beberapa istilah terkait basis data sebagai berikut :

1. *Entity*

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam pada suatu basis data misalnya informasi lalu lintas, *entity* antara lain kemacetan, kecelakaan dan lain sebagainya.

a. *Atribut*

Setiap *Entitas* mempunyai *atribut* atau sebutan untuk mewakili suatu *entity* lalu lintas dengan atributnya, misalnya nama objek, alamat, jenis objek, dan lain sebagainya. *Atribut* juga disebut sebagai data elemen atau data *field*.

b. *Data Value*

Data Value adalah nilai dari data aktual atau informasi yang disimpan pada tiap data elemen atau *atribut*.

2. *Database*

Suatu *database* mendeskripsikan data yang saling berhubungan atau berelasi antara satu dengan yang lainnya. *Database* adalah kumpulan *field-field* yang saling berkaitan sehingga membentuk bangunan data untuk menginformasikan kondisi lalu lintas dalam bahasa tertentu.

3. *File*

Merupakan kumpulan *record-record* sejenis yang mempunyai Panjang elemen yang sama, *atribut* yang sama, namun berbeda-beda datanya.

4. *Record*

Merupakan kumpulan elemen-elemen atau *field* yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap satu *record* mewakili satu data atau informasi

5. *Field*

Suatu *field* menggambarkan suatu *atribut* dari *record*, dimana kumpulan *field* membentuk suatu *record*.

6. *Key*

Adalah sebuah *field* yang digunakan untuk mengidentifikasikan satu atau lebih *atribut* secara unik untuk mengidentifikasi setiap *record*. Terdapat lima jenis *key* yang biasa digunakan, yaitu:

a. *Candidate Key*

Merupakan set atribut minimal yang secara unik mengidentifikasi Setiap kejadian dari sebuah tipe entitas.

b. *Primary Key*

Merupakan *candidate key* yang dipilih untuk mengidentifikasikan setiap kejadian dari suatu tipe *entity* secara unik dan mewakili entitas. *Primary key* atau kunci utama dalam sebuah tabel merupakan kunci yang menjadi identitas dan akan membatasi pengisian *record* dalam sebuah tabel agar tidak duplikat (*redundant*).

c. *Composite Key*

Merupakan sebuah *candidate key* yang terdiri dari dua atau lebih *atribut*.

d. *Foreign Key*

Merupakan sebuah atribut pada suatu relasi yang sama dengan *candidate key* dari relasi lainnya. *Foreign key* atau kunci tamu merupakan kunci yang digunakan sebagai penghubung antara satu tabel dengan tabel lainnya. Jika

sebuah tabel memiliki *foreign key* maka tabel tersebut dapat disebut juga dengan tabel kedua dimana tabel utamanya adalah tabel induk yang memiliki kolom *primary key* dari *foreign key* tersebut.

e. *Alternate Key*

Merupakan kumpulan sebuah *atribut* dari *candidate key* yang tidak terpilih menjadi *primary key*.

7. *Byte*

Atribut dari *field* berupa huruf yang membentuk nilai dari sebuah *field*.

8. *Bit*

Merupakan bagian terkecil dari data secara keseluruhan yaitu berupa karakter ASCII (*American Standard Code Form Information Interchange*).

Suatu *database* dibentuk untuk mengatasi masalah yang sering dihadapi di dalam pengolahan data seperti :

1. Redudansi dan inkonsistensi data

Jika file-file dan program aplikasi diciptakan oleh *programmer* yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada file-file yang berbeda. Penyimpanan data yang sama dan berulang-ulang pada tempat atau media penyimpanan yang berbeda mengakibatkan terjadinya pemborosan media dan inkonsistensi (tidak konsisten).

2. Keamanan data

Setiap pemakai sistem *database* tidak semuanya diperbolehkan untuk mengakses semua data. Misalnya data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan. Keamanan ini dapat diatur lewat program yang dibuat oleh pemrogram atau fasilitas keamanan dari sistem operasi. Dengan *database* manajemen, sistem keamanan data bisa dicapai dengan penggunaan *password* dan wewenang atau *user authorization*.

3. Kesulitan mengakses data

Suatu saat dibutuhkan untuk mencetak data siapa saja, padahal belum

tersedia program yang telah tertulis untuk mengeluarkan data tersebut maka kesulitan tersebut timbul, dan penyelesaiannya untuk itu adalah ke arah Sistem Manajemen Basis Data yang mengambil data secara langsung dengan bahasa yang familiar dan mudah digunakan. *Database* dapat mempermudah pengaksesan data karena mampu mengambil data secara langsung dengan program aplikasi yang mudah digunakan.

4. Isolasi data untuk standarisasi

Jika data tersebar dalam bentuk format yang berbeda-beda, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka suatu *database* haruslah dibuat dalam suatu format yang seragam, sehingga mudah dibuat program aplikasinya.

5. Masalah Integrasi (Kesatuan)

Database berisi file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun diketahui bahwa suatu file berkaitan dengan file lainnya, namun secara teknis maka ada file kunci yang mengaitkan kedua file tersebut.

6. Masalah data *independence* (kebebasan data)

Aplikasi yang dibuat dengan bahasa yang diciptakan dari Sistem Manajemen Basis Data, apapun yang terjadi pada struktur file. Perubahan apapun dalam basis data, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

Relasi tabel merupakan hubungan yang terjadi pada suatu tabel dengan tabel lainnya dengan mempresentasikan hubungan antar objek di dunia nyata, yang berfungsi untuk mengatur operasi suatu basis data. Relasi tersebut menghubungkan antara *primary key* dengan *foreign key*. Hubungan yang dapat dibentuk dapat mencakup 3 (tiga) macam hubungan yaitu:

1. Satu ke Satu (*One to One* / 1-1)

Setiap elemen dari tabel pertama berhubungan paling banyak dengan satu elemen pada tabel kedua. Demikian juga sebaliknya setiap elemen tabel kedua berhubungan paling banyak satu elemen pada tabel pertama. Contohnya : relasi antara tabel mahasiswa dan tabel orang tua. Satu

mahasiswa hanya berhubungan dengan satu orang tua begitu juga sebaliknya.

2. Satu ke Banyak (*One to Many* / 1-M) atau Banyak ke Satu (*Many to One* / M-1)

Setiap elemen dari tabel pertama berhubungan dengan banyak elemen pada tabel kedua. Dan sebaliknya setiap elemen dari tabel kedua berhubungan dengan banyak elemen pada tabel pertama. Contohnya : relasi antara tabel dosen dan tabel mahasiswa. Satu dosen wali dapat berhubungan dengan banyak mahasiswa atau banyak mahasiswa dapat berhubungan dengan satu dosen wali.

3. Banyak ke Banyak (*Many to Many* / M-M)

Banyak elemen dari tabel pertama berhubungan dengan banyak elemen pada tabel kedua demikian sebaliknya. Contohnya : relasi antara tabel mahasiswa dan mata kuliah. Banyak mahasiswa berhubungan dengan banyak mata kuliah.

Agar model *database* relasional bisa digunakan secara efektif, maka pengelompokan yang rumit harus disederhanakan sehingga data berlebihan dan relasi yang salah bisa dieliminasi. Proses membuat struktur data yang lebih kecil dan stabil dari sekelompok data yang rumit disebut normalisasi.

Normalisasi adalah suatu teknik untuk mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai didalam suatu organisasi. Tujuan dari normalisasi:

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.
4. Untuk mengorganisasikan data ke dalam tabel-tabel sehingga memenuhi kebutuhan pemakai.

Proses normalisasi antara lain :

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Jika tabel yang diuji belum memenuhi syarat tertentu maka tabel

tersebut perlu dipecah atau dibagi menjadi beberapa tabel yang lebih sederhana sampai memenuhi data yang optimal.

Bentuk-bentuk dari normalisasi adalah :

1. Bentuk tidak normal (*unformatted form*) : menghilangkan perulangan group. Bentuk ini merupakan bentuk data yang didokumentasikan, tidak ada keharusan untuk mengikuti suatu format tertentu, data bisa saja tidak lengkap atau terduplikasi.

Contoh:

No	Nama	Alamat	No Buku	Judul Buku	Pengarang	Tanggal Pinjam
A001	Rozzy	Jl. Garuda	B001	Ms. Word	Abdul	26/01/2019
A002	Galang	Jl. Merpati	B001	Ms. Word	Abdul	01/02/2019
			B002	Ms. Excel	Widodo	01/02/2019
			B005	Access	Isman	01/02/2019
A003	Fajri	Jl. Elang	B002	Ms. Excel	Widodo	11/02/2019
A004	Edy	Jl. Nuri	B001	Ms. Word	Abdul	01/02/2019
			B005	Access	Isman	18/02/2019
A005	Amos	Jl. Merak	B002	Ms. Excel	Widodo	23/02/2019
			B006	Ms. Visio	Joko	23/02/2019

Gambar 2 4 Gambar contoh tabel bentuk tidak normal

2. Bentuk normal pertama (1NF atau *first normal form*) : menghilangkan ketergantungan sebagian. Bentuk normal pertama mempunyai ciri-ciri yaitu setiap data dibentuk dalam flat file (file dasar) dan data dibentuk dalam satu *record* demi satu *record*. Tidak ada set *atribut* yang berulang-ulang atau atribut yang bernilai ganda, setiap *atribut* dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi).

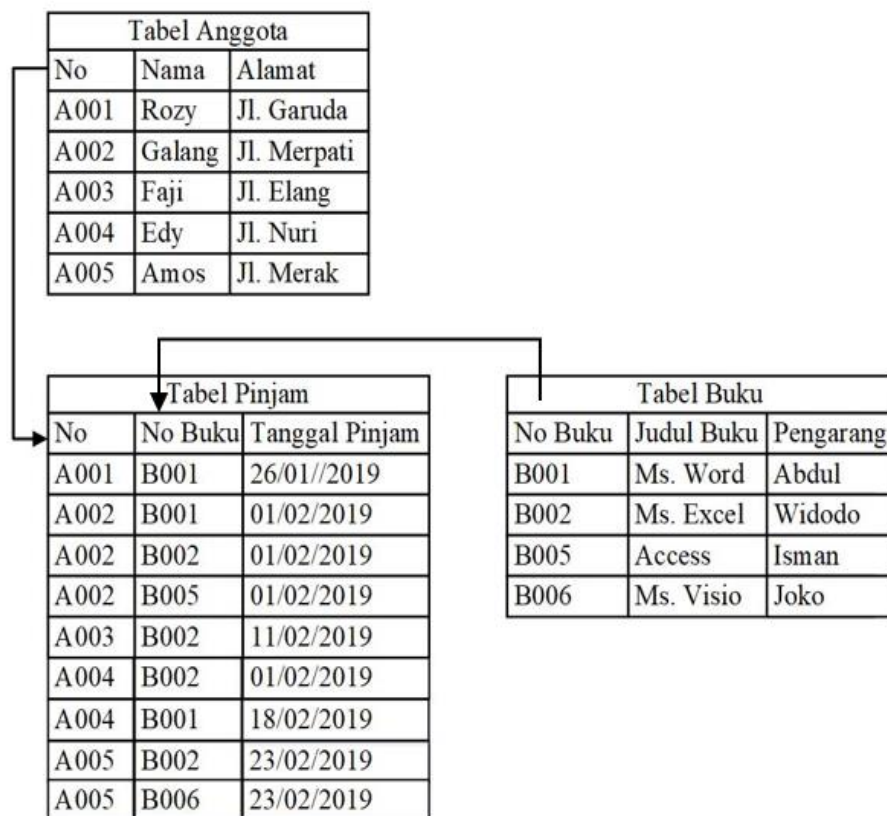
Contoh:

No	Nama	Alamat	No Buku	Judul Buku	Pengarang	Tanggal Pinjam
A001	Rozzy	Jl. Garuda	B001	Ms. Word	Abdul	26/01/2019
A002	Galang	Jl. Merpati	B001	Ms. Word	Abdul	01/02/2019
A002	Galang	Jl. Merpati	B002	Ms. Excel	Widodo	01/02/2019
A002	Galang	Jl. Merpati	B005	Access	Isman	01/02/2019
A003	Fajri	Jl. Elang	B002	Ms. Excel	Widodo	11/02/2019
A004	Edy	Jl. Nuri	B001	Ms. Word	Abdul	01/02/2019
A004	Edy	Jl. Nuri	B005	Access	Isman	18/02/2019
A005	Amos	Jl. Merak	B002	Ms. Excel	Widodo	23/02/2019
A005	Amos	Jl. Merak	B006	Ms. Visio	Joko	23/02/2019

Gambar 2 5 Gambar bentuk tabel normal pertama (1NF)

3. Bentuk normal kedua (2NF atau *second normal form*) : menghilangkan ketergantungan transitif. Bentuk normal kedua memiliki syarat bahwa bentuk data telah memenuhi kriteria bentuk normal pertama, atribut bukan kunci harus bergantung secara fungsi pada kunci utama atau *primary key*, sehingga untuk bentuk normal kedua harus sudah ditentukan kunci- kunci *field*. Kunci *field* harus unik dan dapat mewakili *atribut* lain yang menjadi anggotanya. Jika ada ketergantungan parsial maka *atribut* tersebut harus dipisah pada tabel yang lain, perlu ada tabel penghubung ataupun kehadiran *foreign key* bagi *atribut-atribut* yang telah dipisah tadi.

Contoh:

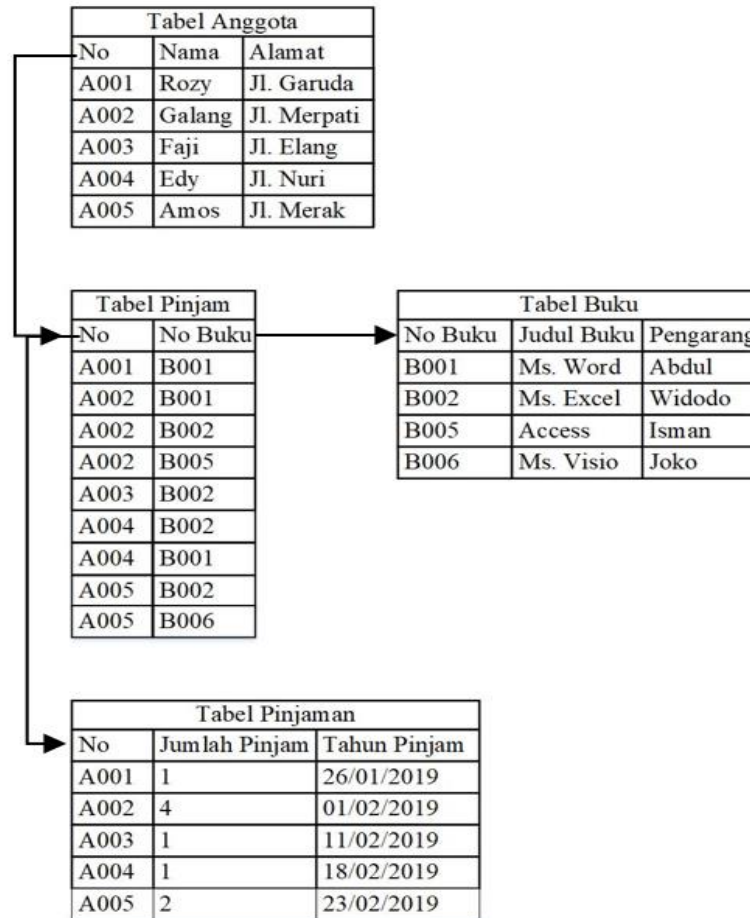


Gambar 2 6 Gambar bentuk tabel normal kedua (2NF)

4. Bentuk normal ketiga (3NF atau *three normal form*) : menghilangkan anomali-anomali hasil dari ketergantungan fungsional.

Untuk menjadi bentuk normal ketiga maka relasi harus dalam bentuk normal kedua dan sama atribut bukan primer tidak punya hubungan yang transitif, dengan kata lain setiap *atribut* bukan kunci harus bergantung pada *primary key* secara menyeluruh (dimana atribut bukan kunci tergantung pada *atribut* bukan kunci lainnya) (Fridayanthie and Mahdiati 2016).

Contoh:



Gambar 2 7 Gambar contoh tabel bentuk normal ketiga (3NF)

2.5.1 MySQL

Dalam mengelola basis data (*database*) perangkat lunak yang sering digunakan ialah MySQL, dimana MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang mempunyai kompatibel dengan berbagai sistem operasi. MySQL adalah sistem yang berguna untuk

melakukan proses pengaturan koleksi-koleksi struktur dan (*database*) baik meliputi proses pembuatan atau proses pengelolaan database. MySQL merupakan software database untuk mengelola dan menyimpan data yang jenisnya beraneka ragam dan tipe data relational yang saling berhubungan. MySQL adalah perangkat lunak sistem manajemen basis data (*database*) SQL atau sering disebut dengan DBMS(*Database Management System*). Penulis menyimpulkan bahwa MySQL merupakan perangkat lunak atau sistem basis data yang saling mempunyai hubungan tipe data relational dalam mengelola data dan menyimpan data.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi *web* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP. Beberapa keunggulan dari MySQL yaitu :

1. Cepat, handal dan mudah dalam penggunaannya MySQL lebih cepat tiga sampai empat kali dari pada database server komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.
2. Didukung oleh berbagai bahasa *Database* server MySQL dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
3. Mampu membuat tabel berukuran sangat besar dan ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
4. Melekatnya integrasi PHP dengan MySQL, keterikatan antara PHP dengan MySQL yang sama-sama software opensource sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan database server lainnya. Modul MySQL di PHP telah dibuat built-in sehingga tidak memerlukan konfigurasi tambahan pada file konfigurasi php ini (Pratiwi, 2020).

2.6 Aplikasi Pendukung

Aplikasi pendukung merupakan aplikasi yang digunakan dalam merancang sistem informasi berbasis komputer.

2.6.1 Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi *editor code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS. Visual Code memudahkan dalam penulisan code yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. Visual Code memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. Visual Studio Code juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di Visual Studio Code (Ramdhan and Devi 2019).

2.6.2 Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya (Riyadli, Arliyana, and Saputra 2020).

2.7 Inventaris

Inventaris adalah kegiatan-kegiatan yang meliputi pendaftaran, pencatatan dalam daftar inventaris, penyusunan atau pengaturan barang-barang milik atau kekayaan negara serta melaporkan pemakaian barang-barang kepada pejabat yang berwenang secara teratur dan tertib menurut ketentuan dan tata cara yang berlaku sehingga mempermudah dalam penyajian data kekayaan negara/pemerintah daerah baik barang-barang tetap maupun barang-barang bergerak. Inventaris merupakan kegiatan melaksanakan pengurusan, penyelenggaraan, pengaturan, pencatatan dan pendaftaran barang inventaris atau hak milik. Hal-hal umum yang diperlukan pada inventaris mencakup:

1. Kode alat/bahan.
2. Nama alat/bahan.
3. Spesifikasi alat/bahan (merek, tipe dan pabrik pembuat alat)
4. Sumber pemberi alat dan tahun pengadaanya
5. Tahun penggunaan
6. Jumlah atau kuantitas
7. Kondisi alat, baik atau rusak.

Manfaat inventaris yaitu:

- a) Mencatat dan menghimpun data aset yang dikuasai unit organisasi/departemen.
- b) Menyiapkan dan menyediakan bahan laporan pertanggung jawaban atas penguasaan dan pengelolaan asset organisasi/Negara.
- c) Menyiapkan dan menyediakan bahan acuan untuk pengawasan aset organisasi atau Negara .
- d) Menyediakan informasi mengenai aset organisasi/Negara yang dikuasai departemen sebagai bahan untuk perencanaan kebutuhan, pengadaan dan pengelolaan perlengkapan departemen.
- e) Menyediakan informasi tentang aset yang dikuasai departemen untuk menunjang perencanaan dan pelaksanaan tugas departemen (Putra 2019).