BAB II
TINJAUAN PUSTAKA

2.1. *Quick Response Code (QR Code)*

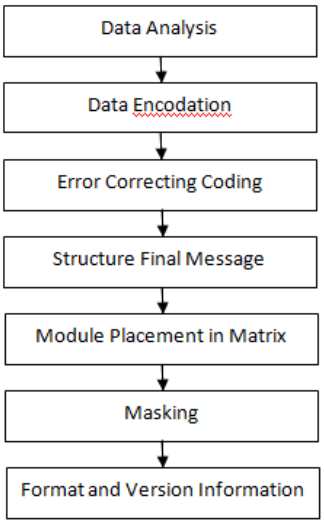
QR Code adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Code* merupakan evolusi dari kode batang (*barcode*) (Nugraha dan Munir, 2011:149). Barcode merupakan gambar garis tegak yang biasanya ditempelkan pada item toko ritel, kartu identitas, dan surat pos untuk mengidentifikasi sejumlah produk tertentu. Secara umum barcode digunakan sebagai UPC (*Universal Price Code*) atau pembaca harga barang secara otomatis. Kode tersebut menggunakan urutan bar vertikal dan spasi untuk mewakili angka dan simbol lainnya (Wahyutama, dkk, 2013). *QR Code* menyimpan informasi secara vertikal dan horizontal (Ciptaningtyas, dkk, 2014 : 71). Berikut contoh penggunaan *QR Code:*

Gambar 2.1. Contoh *QR Code* "Manajemen Informatika Politeknik Negeri Sriwijaya"

QR Code sangat populer karena memiliki kapasitas data yang besar dengan ukuran cetak yang kecil. Kapasitas data yang mampu disimpan oleh QR Code hingga 7.089 karakter numerik, 4.296 karakter alfanumerik, 2.953 *binary bytes*, 1.817 karakater Kanji, jauh lebih tinggi daripada yang lainnya seperti PDF417, DataMatrix dan Maxi Code (Ciptaningtyas, dkk, 2014 : 71).

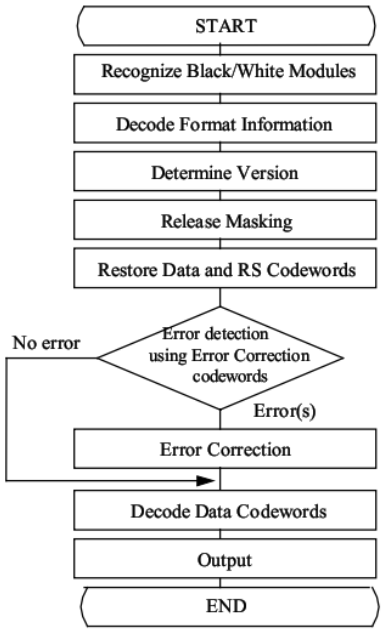
2.1.1. Proses Encoding QR-Code

Prosedur pembangkitan *QR Code* dari sebuah teks dapat dijelaskan dengan diagram alir (Nugraha dan Munir, 2011).



Gambar 2.2. Digram Alir Proses Pembangkit *QR Code*

dan langkah-langkah untuk membaca *QR Code* menjadi teks aslinya merupakan *reverse* atau kebalikan dari langkah-langkah pada pembangkitan *QR Code*. Secara umum prosedur pembacaan *QR Code* dapat dijelaskan dengan diagram alir (Nugraha dan Munir, 2011)



Gambar 2.3. Diagram Alir Proses Pembacaan *QR Code*

2.2. MD5

MD5 (***Message-Digest algortihm 5***) ialah fungsi hash kriptografik yang digunakan secara luas dengan hash *value* 128-bit. Pada standart Internet, MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah berkas ("wiki/MD5", 2018). Pada penelitian yang dilakukan MD5 digunakan untuk menyembunyikan atau mengencode nilai sebelum diconversi ke gambar *Qr Code.* Hal tersebut dilakukan agar tidak dimengerti oleh orang lain.

2.3. JSON

JSON atau *JavaScript Object Notation* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah di- terjemahkan dan dibuat oleh komputer (JSON, 2013). Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C. Oleh karena sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (Ciptaningtyas, dkk, 2014 : 72).

Penggunaan *json* digunakan untuk komunikasi data, perangkat andoid melakukan *request* ke *cloud* lalu mengirimkan ke *server database*, selanjutnya server memberikan response berupa json yang nantinya akan ditangkap android.

2.4. Pemodelan

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka dia akan memodelkannya dengan membuat sebuah maket (tiruan) arsitektur yang akan dibangun dimana maket itu dibuat semirip mungkin dengan desain gedung yang akan dibangun agar artitektur gedung yang diinginkan dapat dilihat (Sukamto dan Shalahudin, 2016:135-136).

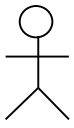
Dalam pembangunan perangkat lunak sistem informasi atau aplikasi juga dibutuhkan pemodelan. Hal ini dilakukan agar mempermudah dapat memberikan

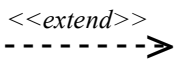
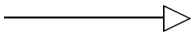
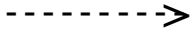
gambaran dalam pembuatan aplikasi, dan dalam pemodelan yang dilakukan antara lain penggunaan diagram seperti *Usecase* diagram, *Activity* Diagram, *Squence* Diagram.

2.4.1. *Usecase* 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 (Sukamto dan Shalahuddin, 2016:155). Adapun simbol-simbol yang digunakan dalam diagram *use case* adalah sebagai berikut :

Tabel 2.1. Simbol-simbol *Usecase* Diagram

No.	Simbol	Deskripsi
1	*Use Case* 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal *frase* nama *use case*.
2	Aktor / *actor* 	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, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3	Asosiansi / a*sosiation* 	Komunikasi antara aktor dan *use case* yang berpartisilpasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

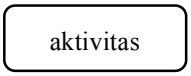
4	Exstensi / *extend* 	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.
5	Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua *buah use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,
6	Menggunakan / *include* / *uses* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

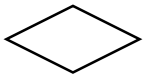
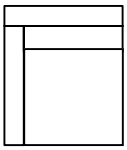
Sumber : Sukamto dan Shalahudin (2016:155-158)

2.4.2. *Activity* Diagram

Diagram aktivitas atau *activity* diagram *diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang. perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Sukamto dan Shalahuddin, 2016:161). Adapun simbol-simbol yang digunakan dalam *activity* diagram adalah sebagai berikut :

Tabel 2.2. Simbol-simbol *Activity* Diagram

No.	Simbol	Deskripsi
1	Status awal / *star* 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas / *activity* 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

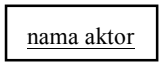
3	Percabangan / *dicision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Pengabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir / *end* 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	*Swimlane* 	Memisahkan organisasi bisnis yang bertanggunga jawab terhadap aktivitas yang terjadi.

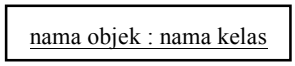
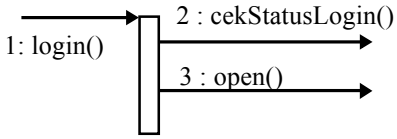
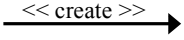
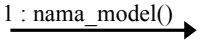
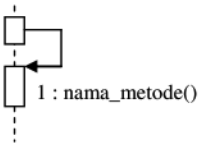
Sumber : Sukamto dan Shalahudin (2016:162-163)

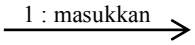
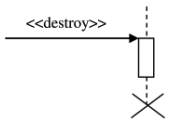
2.4.3. *Squence* Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan massage 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. Membuat diagram sekuen juga di butuhkan untuk melihat skenario yang ada pada *use case* (Sukamto dan Shalahuddin, 2016:165). Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.3. Simbol-simbol *Squence* Diagram

No.	Simbol	Deskripsi
1	Aktor / actor  Atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.

2	Garis hidup / *lifeline* 	Menyatakan objek yang berinteraksi pesan
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5	Pesan tipe *create* 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

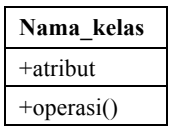
7	Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

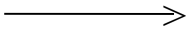
Sumber : Sukamto dan Shalahudin (2016:165-167)

2.4.4. *Class* Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem (Sukamto dan Shalahuddin, 2016:141). Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut :

Tabel 2.4. Simbol-simbol *Class* Diagram

No.	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem
2	Antarmuka / *interface*  **Nama_interface**	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
3	Asosiansi / a*sosiation* 	Ralasi antarakelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*

4	Asosiasi berarah / *direction association* 	Relasi antarakelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai *multiplicity*
5	Generalisasi / *generalization* 	Relasi antarakelas dengan makna generalisasi-spesialisasi (umum khusus)
6	Kebergantungan / *dependency* 	Relasi antarakelas dengan makna kebergantungan antarkelas
7	Agregasi / *aggregation* 	Relasi antarakelas dengan makna semua-bagian (whole-part)

Sumber : Sukamto dan Shalahudin (2016:146-147)

2.5. Database Managemant System (DBMS)

Menurut Sukamto dan Shalahuddin (2016:44). DBMS (*Database Management System*) atau dalam bahasa indonesia sering disebut sebagai Sistem basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data.

Menurut Yanto (2016:16). Database Management System (DBMS) merupakan paket program (software) yang dibuat agar memudahkan dan mengifisienkan pemasukan, pengeditan, penghapusan, dan pengmabilan informasi terhadap database.

Jadi *Database Management System* (DBMS) merupakan suata sistem penyimpanan data dilakukan pengolahan yaitu menambah data, membaca data, mengubah data dan menghapus data.

2.6. Basis Data (*Database*)

Menurut Sukamto dan Shalahuddin (2016:43). Basisdata adalah media untuk untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Dengan demikian dapat diambil kesimpulan bahwa basisdata (*database*) adalah media penyimpanan digital yang menyimpan data pada tabel yang saling berhubungan sehingga dapat diolah dengan mudah dan cepat.

Menurut Priyadi (2014:2). Basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital.

Pada buku kaloborasi SQL & ERD dalam implementasi database, Yudi priyadi (2014:3) menjelaskan beberapa elemen basis data, diantanya adalah sebagai berikut:

1. Tabel

   Pada suatu basis data, tabel dipresentasikan menjadi suatu bentuk segiempat berupa matrik, yang terdiri kolom dan baris. Lokasi pertemuan antara kolom dan baris tersebut memiliki satu nilai tertentu.

2. *Field*

   Kolom merupakan representasi untuk sebuah nama *field* yang pembaca datanya secara vertical. Secara sederhana, *field* dalam suatu tabel dapat dikategorikan menjadi *field key* dan *field non key*.

3. *Record* (baris data)

   Baris merupakan representasi untuk sebuah *record* yang pembaca datanya secara horizoltal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel.

4. Kardinitas

   Karfinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh suatu himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasinya kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri empat macam, yaitu:

   a. Satu ke satu → (1:1).
   b. Satu ke banyak → (1:N).
   c. Banyak ke satu → (N:1).
   d. Banyak ke Banyak → (N:N).

2.7. Pemograman Mobile

2.7.1. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java ("Android_(sistem_operasi)", 2018).

2.7.2. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin ("wiki/Java", 2018).

2.8. Pemograman Website

2.8.1. HTML

Menurut Winarto (2014:1). *Hyper Text Language* (HTML) adalah sebuah bahasa untuk menampilkan konten di *web*.

Menurut Sibero (2014:19). HTML adalah bahasa yang digunakan untuk dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*.

Jadi, HTML adalah bahasa yang digunakan untuk pembuatan sebuah website untuk menampilkan berbagai konten dan juga untuk pertukaran dokumen.

2.8.2. CSS

Menurut Budi (2011:185). *Cascading Style Sheet* (CSS) adalah salah satu bahasa yang bekerja sama dengan dokumen HTML untuk mendifinikan cara bagaimana suatu *web* ditampilkan atau dipersentasikan.

Menurut Prasetyo (2014:185). CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman website (situs).

Jadi, *Cascading Style Sheet* atau biasa dikenal dengan sebutan CSS merupakan bahasa yang digunakan untuk mengatur tampilan dan memperindah tampilan website sehingga *website* tersebut menjadi menarik.

2.8.3. JavaScript

Menurut Prasetyo (2014:292). Javascript adalah program dalam bentuk script, yang dijalan oleh interpreter yang telah ditanamkan kedalam browser web, sehingga browser web dapat mengeksekusi program javascript.

Menurut Budi (2011:221). Javascript adalah bahasa yang berfungsi untuk membuat skrip-skrip program yang dapat dikenal dan dieksekusi oleh web browser dengan tujuan untuk menjadikan halaman web lebih bersifat interaktif.

Jadi, penggunaan javascript bertujuan agar website yang dibuat terlihat lebih interaktif dan menjadi menarik.

2.8.4. PHP *(Hypertext Preprocessor)*

Menerut Sakur (2010:7). PHP merupakan Bahasa pemograman yang disebut sebagai Bahasa scripting, dalam arti PHP merupakan bahasa pemograman

yang ditempel/embedded. Jadi PHP merupakan bahasa scripting yang memiliki tipe interpreter, yang berarti PHP tidak perlu melakukan *compiling*, namun cukup melakukan proses pembacaan pada setiap sintaks yang kemudian melakukan interprestasi dari proses tersebut.

Menurut Supono dan Putratama (2016:3). PHP (PHP: *Hypertext Proccessor*) adalah suatu Bahasa pemograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh computer yang bersifat *server-side* yang dapat ditembahkan kedalam HTML.

Jadi php merupakan sekumpulan *script* web *server-side*, dalam penulisannya dapat dilakukan secara emcedded yaitu mengisipkakan kedalam tag html atau non embedded. Penulisan php dapat dilakukan dengan berbagai cara penulisan, ditulis pada tag <?php … ?> atau ditulis didalam tag *script*. Contoh penulisan PHP seperti berikut:

- Menulis php didalam dokumen html

```
<!DOCTYPE html>
    <html lang="en">
        <head></head>
        <body>
                <?php print “Politeknik Negeri Sriwijaya”; ?>
        </body>
</html>
```

- Menulis php didalam tag script

```
<!DOCTYPE html>
    <html lang="en">
        <head></head>
        <body>
            <script>echo “Politeknik Negeri Sriwijaya”; </script>
        </body>
</html>
```

2.8.5. Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang

untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu ("pengertian-dan-keunggulan-framework-laravel", 2018).

Hal tersebut dapat dilihat dari Contoh penulisan sederhana menggunaan framework laravel sebagai berikut:

Tabel 2.5. Contoh Penulisan Sintak Laravel

No.	Fungsi	Keterangan	Sintak
1	CREATE	Membuat atau mendaftarkan user baru kedalam database yang akan disimpan ke tabel user.	User::*create*(['no_identitas'=> $request->no_identitas, 'nama' => $request->nama, 'email' => $request->email, 'password' => bcrypt ($request->no_identitas)]);
2	READ	Membaca seluruh data pada tabel user.	User::*all*();
3	UPDATE	Mengubah data pada user berdasarkan id user.	User::*update*(['no_identitas'=> $request->no_identitas, 'nama' => $request->nama, 'email' => $request->email])->findOrFail($id);
4	DELETE/ DESTROY	Menghapus data pada tabel user berdasarkan id.	User::*destroy*($id);