



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TELAAH LITERATUR

2.1. Pengertian Gamification

Definisi *gamification* menurut *bunchball* (2005) : “Pada dasarnya, gamification mengaplikasikan hal-hal mekanik game ke dalam sebuah aktifitas non-game yang bertujuan untuk mengubah perilaku seseorang. Jika hal ini diterapkan dalam konteks bisnis maka gamification adalah sebuah proses yang mengintegrasikan game dynamics dan game mechanics kedalam sebuah website, servis bisnis, komunitas online, portal konten atau kampanye marketing yang bertujuan untuk menjangking partisipasi dan engagement.”

Menurut peneliti *Sebastian Deterding* : “gamification adalah pengaplikasian elemen-elemen game design pada konteks non-game” yang terdiri dari 4 komponen:

- *Game*: komponen ini bukan hanya memiliki *playful interaction* tapi juga *playful design*.
- *Element*: bukan sebuah *game* komplit seperti *serious game*.
- *Non-game context*: digunakan untuk tujuan yang berbeda bukan seperti *game* biasa yang bertujuan *entertainment*
- *Design*: *playful design*.



Gambar 2.1 Pengertian *gamification*

Gambar diatas menjelaskan bahwa serious game dan gamification memiliki sedikit perbedaan dimana perbedaan tersebut telah dijelaskan pada poin-poin diatas.

2.2. Partisipasi dan *engagement*

Tujuan dari *gamification* secara keseluruhan adalah untuk mengikat, mengikutsertakan, dan berinteraksi antar sesama konsumen dalam sebuah aktifitas atau komunitas.

2.3. *Game mechanics* dan *game dynamics*

Dua hal diatas sangat berhubungan satu sama lain dan terkadang saling bergantian. *Game mechanics* adalah berbagai mekanisme tindakan, perilaku dan kontrol yang digunakan untuk membuat sebuah aktifitas menjadi lebih menarik. Sedangkan *game dynamics* adalah ketertarikan dari pengalaman yang didapat pada *game mechanics* yang membuat seseorang memiliki hasrat dan motivasi untuk mencapai sesuatu.

2.3.1 *game mechanics* mencakup:

- *Points*
- *Levels*
- *Challenges*
- *Virtual goods*
- *Leaderboards*
- *Gifts and charity*

2.3.2 *game dynamics* meliputi:

- *Reward*
- *Status*
- *Achievement*
- *Self-expression*
- *Competition*
- *Altruism*

2.4 *Arsitektur gamification*

2.4.1 *game mechanics*

Kita mengetahui bahwa *game mechanics* dapat memotivasi perilaku seseorang. Sekarang gamification makin berkembang, *user* dapat memasukan unsur *game mechanics* kedalam aktivitasnya sehari-hari, tapi apa sebenarnya *game mechanics* itu?

Game mechanics adalah alat sekaligus teknik yang digunakan untuk *gamify the website* atau aplikasi. Dengan menggunakan unsur didalam *game mechanics* baik terpisah maupun bersama-sama, memungkinkan *website* atau aplikasi tersebut membangun motivasi *user*. Berikut adalah unsur dalam *game mechanics* antara lain :

- *Points*

Orang-orang menyukai *points*. Mereka suka untuk mengumpulkan *points*, ini membuat *points* merupakan salah satu motivator yang luar biasa. *Points* dapat digunakan sebagai hadiah, *points* juga dapat menunjukkan status *user*, *user* juga dapat menggunakan *point* untuk mengakses konten atau membelanjakannya untuk membeli barang virtual dan hadiah. Sebuah penelitian yang dilakukan IBM dan University of Chicago mengemukakan bahwa efek dramatis dari mengumpulkan *points* dapat mempengaruhi perilaku *user* bahkan ketika *points* hanyalah sekadar *points* dan tidak memiliki nilai asli. Orang-orang hanya suka mengumpulkannya dan merasa seperti mereka mendapatkan sesuatu.

- *Levels*

Level layaknya perbedaan kelas pada frequent flyer program atau warna sabuk pada ilmu bela diri atau jabatan dalam sebuah perusahaan, level adalah sebuah indikasi bahwa *user* telah mendapat sebuah pencapaian. Level sering didefinisikan sebagai awal mula *points*, *user* yang sering berpartisipasi akan cepat naik level dan menggunakan level tersebut untuk mengatur atau mengakses konten dalam sebuah situs.

- *Challenges, trophies, badges, achievements*

Challenges atau tantangan memberikan misi kepada *user* untuk diselesaikan dan memberikan hadiah ketika tantangan tersebut telah diselesaikan. Tantangan memberikan sebuah *goal* dan perasaan bahwa *user* sedang mengejar sesuatu. Hadiah yang diberikan berupa *trophies, badges* atau *ribbon*. *Trophies, badges* atau *ribbon* tersebut merupakan indikator bahwa *user* telah menyelesaikan tantangan tersebut. Cara agar *levels* dan *achievement* tersebut lebih menarik adalah dengan menyediakan tempat untuk “memamerkan” *trophies* atau *badges* tersebut, sama seperti di dunia asli jika seseorang memiliki warna yg berbeda pada kartu kredit menunjukkan limit kartu kredit atau warna kartu yang berbeda pada *frequent flyer program* menunjukkan seberapa sering orang tersebut menggunakan jasa maskapai penerbangan tersebut.

- *Virtual goods*

Virtual goods bukan merupakan barang yang *real* yang dapat dibeli dari sebuah komunitas *online* atau *game online*. User membeli barang seperti pakaian, senjata atau dekorasi untuk membangun identitas selagi user membandingkan atau “memamerkannya” kepada teman-teman. *Virtual goods* juga dapat digunakan sebagai sumber pendapatan dengan cara menjualnya dengan uang asli.

- *Leaderboards*

Kebanyakan *game* yang sukses pasti memasukkan unsur *leaderboard* atau *high score table* kedalam gamenya. *Leaderboards* membawa aspirasi, popularitas dan *highlight*. Dalam konteks *gamification*, *leaderboards* digunakan untuk berkompetisi dengan sesama *user*.

- *Competition*

Kompetisi memungkinkan *user* untuk menantang satu sama lain untuk mendapatkan skor tertinggi dari suatu aktivitas. Begitu suatu aktivitas selesai, *user* dengan skor tertinggi akan mendapatkan hadiah utama sedangkan yang lain akan mendapatkan hadiah hiburan. Kompetisi akan menjadi sangat kompetitif jika terdapat salah seorang *user* mempost di social media bahwa Ia telah mencapai suatu tujuan misalnya “*I just score 500.000 points at monkey escape, I dare you to beat that!*”.

2.4.2 *Game dynamics*

Kenapa orang termotivasi oleh *game mechanics*? Jawabannya adalah karena orang memiliki kebutuhan dasar dan hasrat, hasrat untuk mendapatkan hadiah, status, pencapaian, ekspresi diri, kompetisi dan *altruism*. Kebutuhan ini bersifat universal dan lintas generasi, wilayah, budaya, dan jenis kelamin. *Game designer* telah mengetahui hal ini selama bertahun-tahun dalam aplikasinya didalam sebuah *game*, dan

gamification sekarang memungkinkan pedoman ini diaplikasikan lebih luas. Dengan mengemas unsur *game mechanics* kedalam *website*, aplikasi dan komunitas, kita dapat menciptakan pengalaman yang dapat mengubah perilaku dengan memenuhi satu atau lebih kebutuhan tersebut. Unsur *game dynamics* antara lain:

- *Reward*

Manusia sangat termotivasi jika ada hadiah didalam setiap tindakan yang dilakukan. Sebuah hadiah baik nyata ataupun tidak, diberikan ketika *user* telah melakukan suatu tindakan, hadiah juga memicu *user* tersebut untuk menjalankan aktivitas tersebut lagi. Dengan *gamification*, hadiah bisa didapatkan dengan cara mengumpulkan *point* atau sejenisnya dan ini merupakan salah satu cara untuk memuaskan hasrat manusia.

- *Status*

Kebanyakan manusia menginginkan atau membutuhkan status, pengakuan, popularitas, martabat, perhatian dan terutama penghargaan dan hormat dari orang lain. Manusia membutuhkan aktivitas dalam rangka mendapatkan semua hal tersebut. Elemen-elemen dalam *game mechanics* seperti misalnya *leveling-up* dari kartu kredit *gold* menjadi kartu kredit *platinum* menjadi motivator utama.

- *Achievement*

Beberapa orang termotivasi oleh kebutuhan atau hasrat untuk mendapatkan, untuk menyelesaikan sesuatu melalui usahanya, untuk mencapai tujuan, dan untuk menang. Orang-orang termotivasi oleh pencapaian dan cenderung untuk mencari tantangan yang sulit namun dapat diselesaikan.

- *Self-expression*

Banyak orang ingin dan butuh kesempatan untuk mengekspresikan diri mereka masing-masing. Hal ini berkaitan dengan hasrat manusia untuk menunjukkan gaya, identitas, dan kepribadian dan untuk menunjukkan relasinya dengan sebuah kelompok. *Virtual goods* merupakan jalan yang sesuai jika seseorang ingin mengekspresikan dirinya. Tidak peduli apakah *user* meraihnya melalui *reward*, menerima hadiah maupun membeli langsung dengan kurs asli. Avatar seseorang biasanya merupakan ekspresi dirinya.

- *Competition*

Orang dapat termotivasi melalui kompetisi. Ini telah dibuktikan bahwa performa yang baik dapat diraih jika ada lingkungan yang kompetitif dan pemenangnya mendapatkan hadiah. Ini terjadi karena orang dapat membandingkan hasil kerjanya dengan orang lain. Semua elemen dalam *game mechanics* dapat mempengaruhi kompetisi namun kompetisi tidak bisa lepas dari *leaderboards*. Hampir semua *games* menampilkan setidaknya top 10 list.

- *Altruism*

Altruism atau gift giving adalah motivator yang baik jika kita memiliki komunitas dimana orang-orang mencari relasi. Dalam *gamification*, pemberian hadiah merupakan motivator yang luar biasa. *User* dapat mendapatkan hadiah dari *user* lain dan menarik *user* lain kedalam *game* dan setiap *user* tersebut menerima sebuah hadiah, hadiah tersebut membuat *user* tersebut mengakses *game* itu lagi untuk mengambil hadiahnya, dan proses ini akan terus terulang dan menciptakan suatu keterikatan.

Game Mechanics	Human Desires					
	Reward	Status	Achievement	Self Expression	Competition	Altruism
Points	●	●	●		●	●
Levels		●	●		●	
Challenges	●	●	●	●	●	●
Virtual Goods	●	●	●	●	●	
Leaderboards		●	●		●	●
Gifts & Charity		●	●		●	●

gambar 2.2 hubungan hasrat manusia dengan game mechanics dan game dynamics

2.5 Games are everywhere

Manusia telah bermain banyak jenis game mulai dari jaman batu, dan kompetisi telah ada sejak saat itu. Di jaman modern seperti sekarang, dengan banyaknya waktu luang yang dimiliki manusia, dan game merupakan industry yang berkembang dan sangat menguntungkan dengan 60 milyar dollar atau setara dengan kurang lebih 540 trilyun rupiah per tahun. Hasilnya gamification menjadi sebuah perantara yang baik untuk mengajarkan, meyakinkan dan memotivasi orang. Banyak aktivitas yang orang lakukan dan terkandung unsur game mechanics didalamnya namun orang-orang tidak menyadarinya. Berikut beberapa contoh aktivitasnya:

- **Frequent Flyer Programs**

Lebih dari 120 juta orang di seluruh dunia mengikuti program ini, alurnya sederhana mengumpulkan point, naik level kemudian mendapatkan hadiah. Program ini ditawarkan oleh hamper seluruh maskapai penerbangan besar, maksudnya tidak lebih dari menjaring komunitas dan “mengikat” mereka. Program ini sebenarnya adalah sebuah game dengan cara pelanggan mendapatkan miles (points) dari setiap kali pelanggan tersebut terbang kemudian naik level atau status dan jika pelanggan tersebut terbang menggunakan jasa maskapai yang sama dalam 90 hari kedepan maka pelanggan tersebut akan mendapatkan bonus 2500 miles. Dengan cara tersebut, banyak orang setia terhadap maskapai penerbangan tersebut bahkan ketika pelayanannya menurun atau jelek saat itu.

- **Starbucks dan Foursquare**

Starbucks, salah satu coffee shop terbesar didunia menggunakan Foursquare untuk menjaring komunitasnya. Foursquare adalah sebuah contoh gamification dimana seseorang dapat melakukan check-in dan mendapatkan virtual reward. Dengan menggunakan Foursquare, orang-orang dapat melakukan check-in di semua gerai Starbucks melalui handphone dan mendapatkan points dan menyelesaikan quest seperti mengunjungi 5 gerai Starbucks berbeda untuk mendapatkan badges. Points yang didapat tidak memiliki nilai komersil dan badges juga tidak memiliki nilai yang real. Namun cara ini terbukti efektif dalam meningkatkan penjualan Starbucks.

- **Nike+ dan iPod**

Nike, perusahaan manufaktur terkemuka didunia telah memasukan unsur gamification kedalam produknya dan meluncurkan aplikasi Nike+ pada tahun 2008. Lebih dari 1,8 juta orang menggunakannya, Nike+ dirancang untuk menghitung data seperti jarak yang telah ditempuh, langkah dan berapa kalori yang telah dibakar menggunakan GPS sensor didalam iPod dan memberikan reward jika pelari tersebut mencapai target yang ditentukan.

2.6. Sains dibalik *Game dan Gamification*

Para peneliti dari bidang psikologis, industri game dan akademik telah belajar bagaimana game dapat memotivasi psikologis seseorang. Professor psikologis, Mihaly Csikszentmihalyi menjabarkan bahwa setidaknya ada 4 kondisi dan 3 karakteristik dalam sebuah game yang dapat membantu player mendapatkan kepuasan sebagaimana terlihat pada tabel 2.1 dan 2.2.

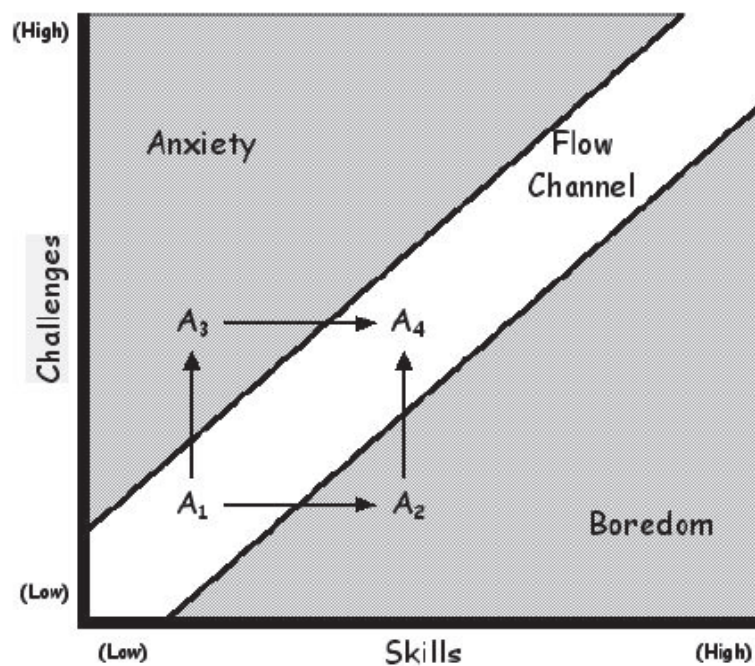
Tabel 2.1 Tabel kondisi

Kondisi	Penjelasan
Tugas yang jelas	Player mengerti tugas apa yang harus diselesaikan
Focus	Player tidak tertarik dengan hal lain dan hanya focus kepada tugas yang ada
Tujuan yang jelas	Goals yang menantang dan masi dalam kapasitas player
Feedback	Player menerima notification tugas mana yang selesai dan tugas mana yang belum selesai

Tabel 2.2 Tabel karakteristik

Karakteristik	Penjelasan
Kontrol	Player percaya bahwa tindakan yang dilakukan berdampak langsung terhadap game
Berkurangnya kesadaran diri	Player sangat focus dengan tugas yang diberikan sehingga tidak merasakan apa-apa bahkan hanya untuk ke toilet
Lupa waktu	Player tidak lagi mengenal waktu

Untuk mendapatkan flow yang benar, maka dibutuhkan kondisi yang benar, kondisi yang terakhir dan paling penting adalah menyeimbangkan antara *goal* yang menantang namun dapat diraih dengan kemampuan user atau player. Sebuah tugas yang tidak menantang atau membutuhkan waktu panjang akan menjadi sangat membosankan dan player akan kehilangan minatnya. Dengan meningkatnya kemampuan player harus juga diimbangi dengan tantangan yang semakin sulit. Keseimbangan ini mengacu pada *flow* Czikszent mihalyi berikut :

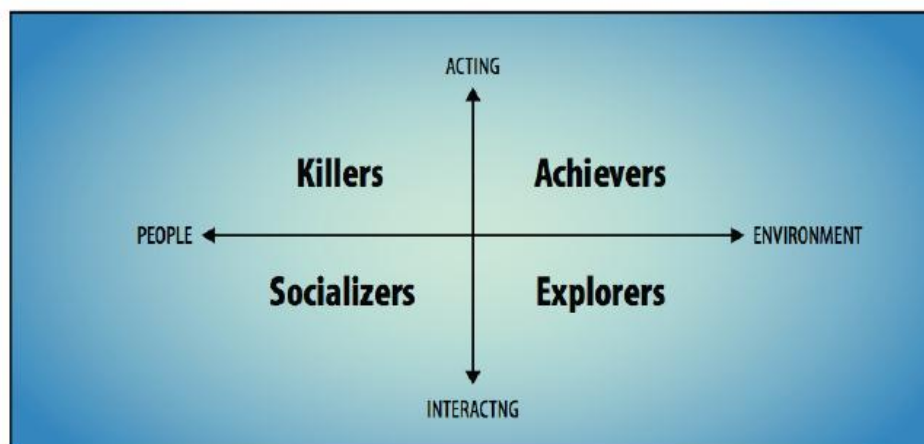


Gambar 2.3 Flow: The psychology of Optimal Experience

Flow diatas menggambarkan keadaan psikologis seseorang dimana A1 merupakan psikologis seseorang pada saat memulai suatu *game* yang kemudian akan menuju ke A2 dan A3 dimana A2 memerlukan kemampuan pemain dalam menyelesaikan suatu misi dan A3 merupakan tantangan yang harus diselesaikan oleh pemain, dibutuhkan keseimbangan agar pemain tidak merasa bosan dan kebingungan yang ditunjukkan pada A4.

2.7. Tipe Pemain

Untuk lebih memahami mengapa orang-orang bermain *game*, Richard Bartle mengidentifikasi empat kepribadian pemain dengan mempelajari pemain-pemain MMOG (Massively multiplayer online game), berikut gambarnya:

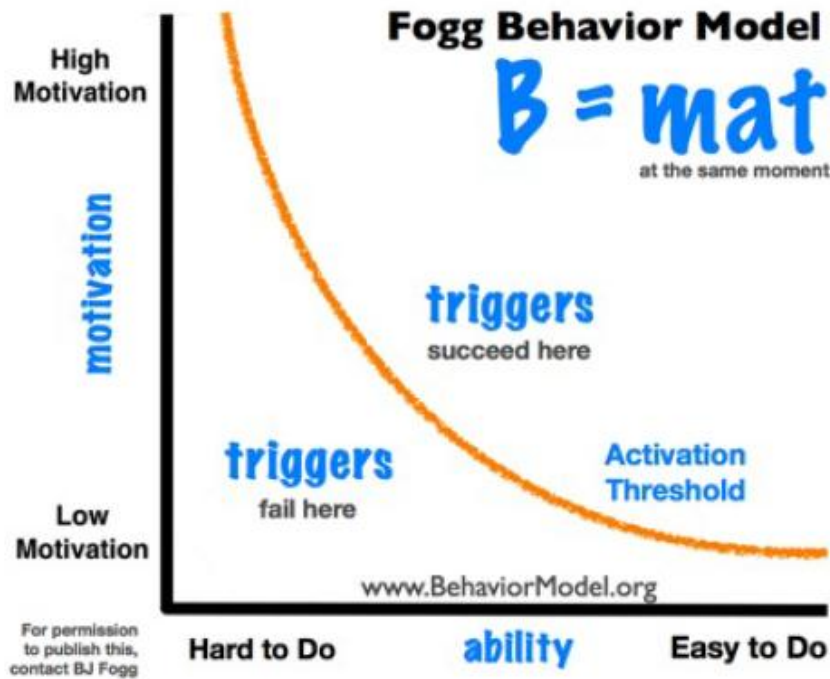


Gambar 2.4 Bartle's Player Type

- *Explorers* memiliki kecenderungan atau ketertarikan pada penemuan
- *Achievers* sangat termotivasi pada akumulasi *points, status dan ranking*.
- *Killers* termotivasi pada tantangan, kompetisi dan kecepatan dalam setiap tugas.
- *Socializers* menyukai sosialisasi atau interaksi dengan sesama. Sebuah studi mengemukakan bahwa kira-kira 80% pemain adalah *socializers* dan sisanya *explorers, achievers dan killers* hanya 20% dari populasi pemain.

2.8 Fogg Behavior Model

Peneliti dari Stanford University BJ Fogg mengenalkan Fogg Behavior Model (FBM) untuk menjelaskan bagaimana perilaku seseorang berubah. Dalam model tersebut terlihat ada tiga elemen, motivasi, *ability dan trigger* yang berjalan bersamaan.



Gambar 2.5 Fogg Behavior Model

- *Motivasi*: seseorang yang sangat ingin melakukan sesuatu
- *Ability*: seseorang dengan mudah dapat menyelesaikan aktivitas
- *Trigger*: seseorang yang memicu terjadinya suatu kejadian

Michael Wu, seorang peneliti *gamification* pernah menggunakan FBM untuk menganalisa mengapa dan bagaimana *gamification* dapat memicu aktivitas. Michael Wu berkata “*Game mechanics dan game dynamics* dapat dengan positif memicu terjadinya perubahan perilaku pada seseorang dan memicu tindakan yang spesifik. Dengan kata lain, *gamification* dapat berjalan sukses jika ketiga hal diatas berjalan bersama.”

2.9 Barcode

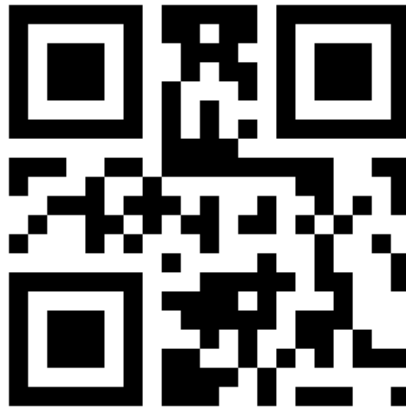
Menurut The Columbia Electronic Encyclopedia:

“Barcode adalah sebuah sistem pengkodean komputer yang menggunakan sebuah pola yang tersusun atas garis atau balok untuk mengidentifikasi produk, surat dan paket, dan sebagainya. Bar codes dibaca melalui scan pada barcode tersebut dengan menggunakan sebuah program komputer untuk menerjemahkannya.”

Konsep awal barcode adalah dengan menggunakan kumpulan bars atau balok yang biasanya berwarna hitam dan putih yang tersusun dalam sebuah garis lurus, melalui scan dan interpretasi dari pola tersebut maka akan menghasilkan informasi yang terkandung dalam barcode tersebut. Walaupun sangat dapat diandalkan, barcodes 1 dimensi ini hanya dapat menyimpan informasi dalam jumlah yang sangat kecil, biasanya hanya sampai 13 digit alphanumeric. Keterbatasan penyimpanan informasi tidak dapat mengakomodir luasnya pemakaian barcode 1 dimensi ini sehingga diciptakanlah sebuah teknologi pengembangan dari barcode 1 dimensi ini yaitu barcode 2 dimensi dalam bentuk kode matriks.

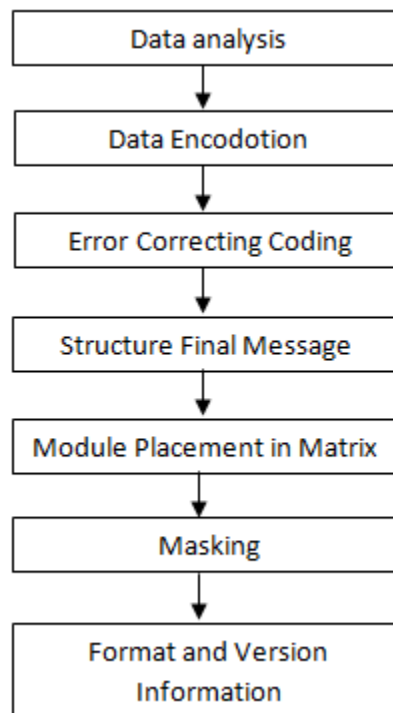
2.10 QR Code

Adalah image berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data didalamnya. QR Code merupakan evolusi dari Barcode. Barcode merupakan simbol yang tersusun dari pola batang-batang berwarna hitam dan putih yang dapat dikenali oleh komputer. Berikut adalah contoh dari sebuah QR Code :



Gambar 2.6 contoh gambar *QR Code*

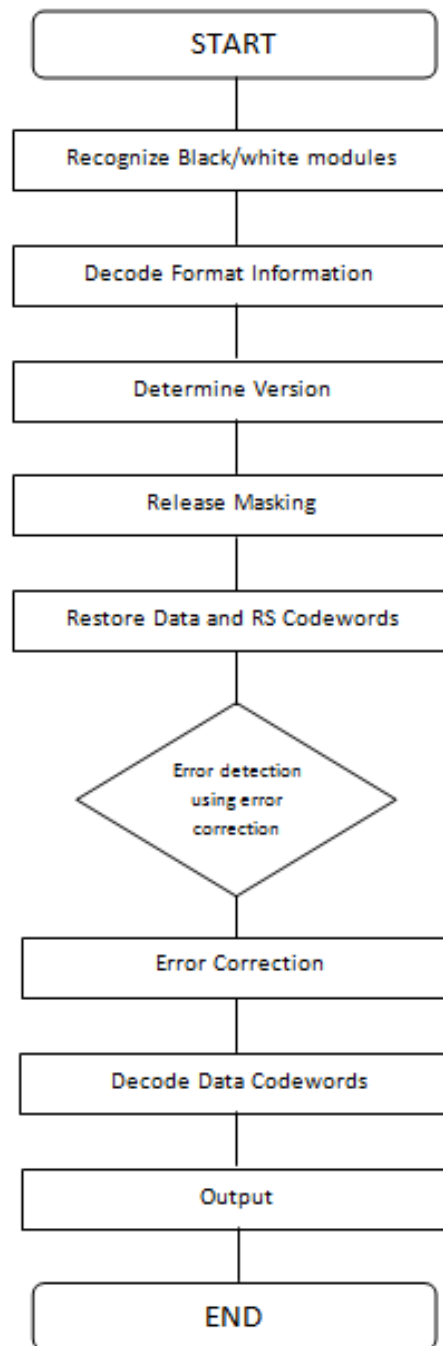
QR Code merupakan singkatan dari *Quick Response Code*. *QR Code* dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang bergerak di bidang otomotif. *QR Code* ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula. Prosedur pembangkitan *QR Code* dari sebuah teks dapat dijelaskan dengan diagram alir pada gambar berikut.



Gambar 2.7 Diagram Alir proses pembangkitan QR Code

Prosesnya di mulai dengan analisa data yang ingin dijadikan *QR Code*, kemudian sistem akan melakukan *encode* data yang kemudian menjalankan algoritma *error correction* untuk mendapatkan *structure final message* lalu sistem akan melakukan *module placement* dalam bentuk matriks dan melakukan *masking* pada *QR Code* yang akan membentuk format dan *version information*.

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 pada gambar berikut.



Gambar 2.8 Diagram alir proses pembacaan *QR Code*

Gambar di atas merupakan diagram alir pembacaan QR Code yang dimulai saat sistem mengenali black or white modlues yang kemudian sistem akan melakukan decode pada format information, lalu sistem akan menerjemahkan versi dari QR Code yang di baca serta menjalankan algoritma release masking yang berguna untuk membaca informasi yang ada pada QR Code kemudian sistem akan melakukan restrore data selanjutnya sistem akan melakukan deteksi error dengan menggunakan algoritma error correction untuk melakukan decode pada data yang kemudian akan menampilkan output dari QR Code yang di scan.

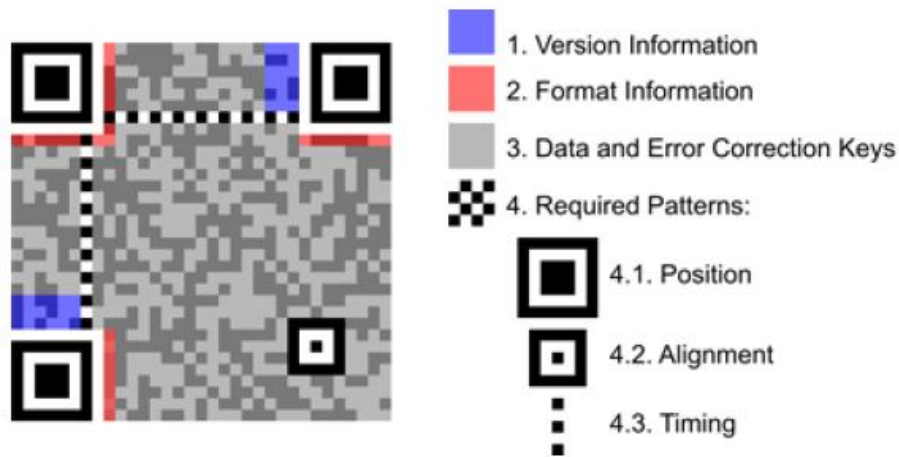
Perbandingan barcode 1 dimensi dengan barcode 2 (*QR Code*) dimensi:



Gambar 2.9 perbandingan barcode 1 dimensi dan 2 dimensi

Dapat dilihat dari gambar diatas, *QR Code* mengandung informasi baik di bagian vertikal maupun bagian horizontal, dimana barcode 1 dimensi hanya mengandung informasi di satu arah. Dengan model ini, *QR Code* dapat menampung informasi yang lebih banyak dibandingkan dengan barcode 1 dimensi.

2.10.1 Struktur *QR Code*



Gambar 2.10 struktur *QR Code*

Sebuah *QR Code* terdiri dari beberapa bagian:

- *Version information* – informasi mengenai *QR Code* itu sendiri, versi, dan ukuran.
- *Format information* – informasi mengenai level koreksi dan pola mask.
- *Data and error correction keys* – adalah tempat dimana data yang sudah diencode tersimpan.
- *Alignment pattern* – digunakan untuk menemukan posisi yang tepat saat terjadi *decoding*
- *Timing pattern* – digunakan untuk menentukan koordinat symbol dalam aplikasi *decoder*

2.10.2 Versi *QR Code*

Versi dari sebuah *QR Code* ditentukan dari ukuran *QR Code* itu sendiri. *QR Code* paling kecil yaitu 21 x 21 matriks atau yang lebih dikenal dengan versi 1. Setiap versi memiliki perbedaan sebanyak empat modul, jadi versi 2 adalah 25 x 25 matriks, versi 3 adalah 29 x 29 matriks dan seterusnya. Sampai saat ini ada 40 versi *QR Code* yang tersedia, dan yang terbesar adalah 177 x 177 matriks.

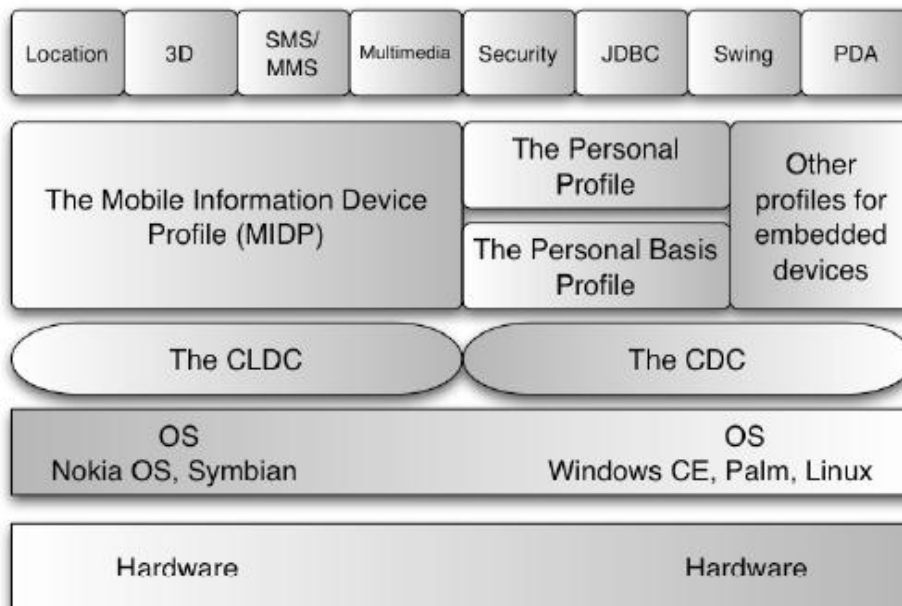
2.10.3 *QR Code error-correction*

Kapasitas data dari sebuah *QR Code* juga ditentukan dari kemampuan *error-correction*-nya. Terdapat empat level kemampuan *error correction* dalam *QR Code*:

- Level L : sekitar 7% atau kurang *error* yang dapat diperbaiki
- Level M : sekitar 15% atau kurang *error* yang diperbaiki
- Level Q : sekitar 25% atau kurang *error* yang dapat diperbaiki
- Level H : sekitar 30% atau kurang *error* yang dapat diperbaiki

2.11 Java Platform Micro Edition (Java ME)

Java platform micro edition atau yang lebih dikenal dengan J2ME menyediakan fleksibilitas dalam menjalankan dan membangun aplikasi mobile pada perangkat mobile seperti telepon genggam, PDA, dan perangkat mobile lainnya. Java ME memperbolehkan *developers* untuk menciptakan *multiplatform wireless application* yang dapat diimplementasikan kedalam berbagai telepon genggam yang mendukung teknologi java.



Gambar 2.11 struktur java ME

Sun java wireless toolkit untuk CLDC atau yang lebih dikenal dengan java 2 platform, micro edition (J2ME) wireless toolkit adalah sebuah state-of-the-art toolbox untuk membangun aplikasi wireless yang berbasis java ME *Connected Limited Device Configuration* (CLDC) dan *Mobile Information Device Profile* (MDIP), dan dirancang agar dapat dijalankan didalam telepon genggam, PDA dan perangkat mobile lainnya.

2.11.1 *Connected Limited Device Configuration (CLDC)*

CLDC digunakan untuk membangun java wireless application dengan MDIP. CLDC diimplementasikan pada perangkat keras dengan keterbatasan memory, dari 160kb sampai 512kb. Beberapa fitur yang tidak begitu penting dari java 2 platform yaitu *standard edition* (J2SE) dihapus berdasarkan keterbatasan ini.

2.12 MIDlets

Aplikasi yang berjalan di sebuah handheld *device* yang mendukung MIDP disebut MIDlets. MIDlet adalah sebuah aplikasi berbasis J2ME dan MIDP profile. Seperti program java lainnya, MIDlet dapat dijalankan diberbagai mobile device. MIDP biasanya digunakan untuk membangun aplikasi pada handset yang memiliki keterbatasan CPU seperti memori yang rendah, *keyboard* sederhana dan layar yang kecil.

2.13 Zebra Crossing (ZXing)

ZXing adalah sebuah *open-source, multi format barcode 1D/2D barcode reader library* yang diimplementasi dalam java. Tujuan dari ZXing ini adalah untuk membantu proses *decoding* dari sebuah QR Code, data matriks, dan UCP dari 1D barcodes.

2.14 Unified Modeling Language (UML)

Unified Modeling Language adalah sebuah bahasa yang diterima dan digunakan oleh software developer dan software analis sebagai suatu bahasa yang cocok untuk merepresentasikan grafik dari suatu relasi antar entitas-entitas software (Gornik, 2003). Dengan menggunakan UML, tim pengembang software akan mempunyai banyak keuntungan, seperti memudahkan komunikasi dengan sesama anggota tim tentang software apa yang akan dibuat, memudahkan integrasi ke dalam area pengerjaan software karena bahasa ini berbasiskan meta-models dimana meta-models bisa mendefinisikan proses-proses untuk mengkonstruksikan konsep-konsep yang ada. UML juga menggunakan format input dan output yang sudah mempunyai bentuk standar yaitu XML Metadata Interchange (XMI), menggunakan aplikasi dan pemodelan data yang universal, merepresentasikan dari tahap analisis ke implementasi lalu ke deployment yang terpadu, dan mendeskripsikan keutuhan tentang spesifikasi software (Hofmeister, Nord, dan Soni, 1999).

UML menyediakan kumpulan alat yang sudah terstandarisasi, yang digunakan untuk mendokumentasikan analisis dan perancangan sebuah sistem perangkat lunak. (Kendall & Kendall, 2005, p663) Peralatan utama UML adalah diagram-diagram yang digunakan untuk membantu manusia dalam memvisualisasikan proses pengembangan sebuah sistem perangkat lunak, sama seperti penggunaan denah (*blueprint*) dalam pembuatan bangunan.

2.14.1 Use Case Model

Use case model adalah teknik pemodelan untuk mendapatkan *functional requirement* dari sebuah sistem, menggambarkan interaksi antara pengguna dan sistem, menjelaskan secara naratif bagaimana sistem akan digunakan, menggunakan skenario untuk menjelaskan setiap aktivitas yang mungkin terjadi. Ada beberapa bagian didalam *use case model* (Dharwiyanti dan Wahono, 2003).

- *Use Case*, untuk mengetahui *action* atau prosedur apa yang ada didalam sistem.



Gambar 2.12 *use case* pada *use case* diagram

- *Actor*, siapa saja yang terlibat dalam action tersebut.



Gambar 2.13 *actor* pada use case diagram

2.14.2 Class diagram

Class Diagram merupakan diagram paling umum yang dijumpai dalam pemodelan berbasis UML. Didalam Class Diagram terdapat class dan interface beserta atribut-atribut dan operasinya, relasi yang terjadi antar objek, constraint terhadap objek-objek yang saling berhubungan dan inheritance untuk organisasi class yang lebih baik. Class Diagram juga terdapat *static view* dari elemen pembangun sistem. Pada intinya Class Diagram mampu membantu proses pembuatan sistem dengan memanfaatkan konsep forward ataupun reverse engineering (Rational Software Corporation, 1997).

Class diagram memiliki 2 komponen penting, yaitu:

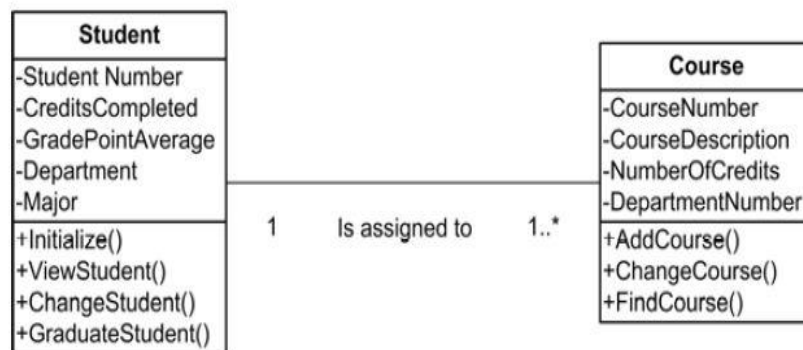
- *Structural*, yaitu ciri pembeda objek.
- *Behavioral*, yaitu tingkah laku atau kegiatan yang mampu dilakukan oleh objek.

Class Diagram menggambarkan karakteristik statis sistem tanpa menjelaskan proses secara mendetil. Sebuah class diagram juga memperlihatkan hubungan antar *class*. (Kendall & Kendall, 2005, p678)

Berbagai symbol yang hadir didalam class diagram antara lain adalah:

- a. *Class*, yang berfungsi untuk merepresentasikan tipe dari data yang dimilikinya. Class diagram dapat ditampilkan dengan menunjukkan atribut dan operasi yang dimilikinya atau hanya menunjukkan nama *class*-nya saja. Dapat juga kita tuliskan nama *class* dengan atributnya saja atau nama class dengan operasinya.
- b. *Attribute*, merupakan data yang terdapat didalam *class* dan *instance*-nya dengan operator.
- c. *Operation*, berfungsi untuk merepresentasikan fungsi-fungsi yang ditampilkan oleh class dan *instance*-nya dengan operator.
- d. *Association*, digunakan untuk menunjukkan bagaimana dua *class* berhubungan satu sama lainnya. *Association* ditunjukkan dengan sebuah garis yang terletak diantara dua *class*. Didalam setiap *association* terdapat *multiplicity*, yaitu simbol yang mengindikasikan berapa banyak *instance* dari *class* pada ujung *association* yang satu dengan *instance class* di ujung *association* lainnya.
- e. *Generalizations*, berfungsi untuk mengelompokkan *class* ke dalam hirarki inheritance.
- f. *Aggregation*, merupakan bentuk khusus dari *association* yang merepresentasikan hubungan “*part-whole*”. Bagian “*whole*” dari hubungan ini sering disebut dengan assembly atau *aggregate*. *Class* yang satu dapat dikatakan merupakan bagian dari *class* yang lain yang ikut membentuk *class* tersebut.

- g. *Composition*, merupakan jenis *aggregation* yang lebih kuat diantara dua *class* yang memiliki association dimana jika *whole* ditiadakan, maka *part*-nya juga ikut ditiadakan. Berbeda dengan *aggregation*, *part* akan tetap bisa berdiri sendiri meskipun bagian *whole*-nya ditiadakan.
- h. Penggunaan operator (+) dalam class diagram diartikan dengan *public*, operator (-) diartikan *private*, dan operator (#) diartikan *protected*.



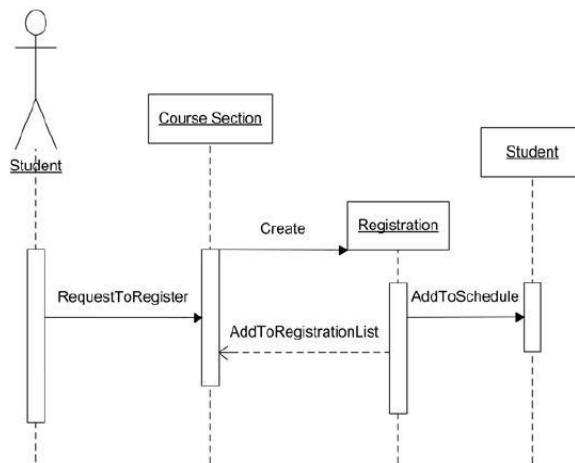
Gambar 2.14 contoh class diagram (Kendall & Kendall, 2005, p685)

2.14.3 Sequence diagram

Menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau behaviour yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma (Hofmeister, Nord, & Soni, 1999).

Dalam sequence diagram terdapat beberapa bagian.

- a. *Participant*, yaitu objek yang terkait dengan sebuah urutan proses.
- b. *Lifeline*, menggambarkan daur hidup sebuah objek.
- c. *Activation*, suatu titik waktu dimana sebuah objek mulai berpartisipasi dalam sebuah *sequence*.
- d. *Time*, elemen paling penting dalam sequence diagram yang konteksnya adalah urutan, bukan durasi.
- e. *Return*, suatu hasil kembalian sebuah operasi. Operasi mengembalikan hasil tetapi boleh tidak ditulis jika tidak ada perbedaan dengan Getter-nya.



Gambar 2.15 Contoh Sequence Diagram (Kendall & Kendall, 2005, p682)

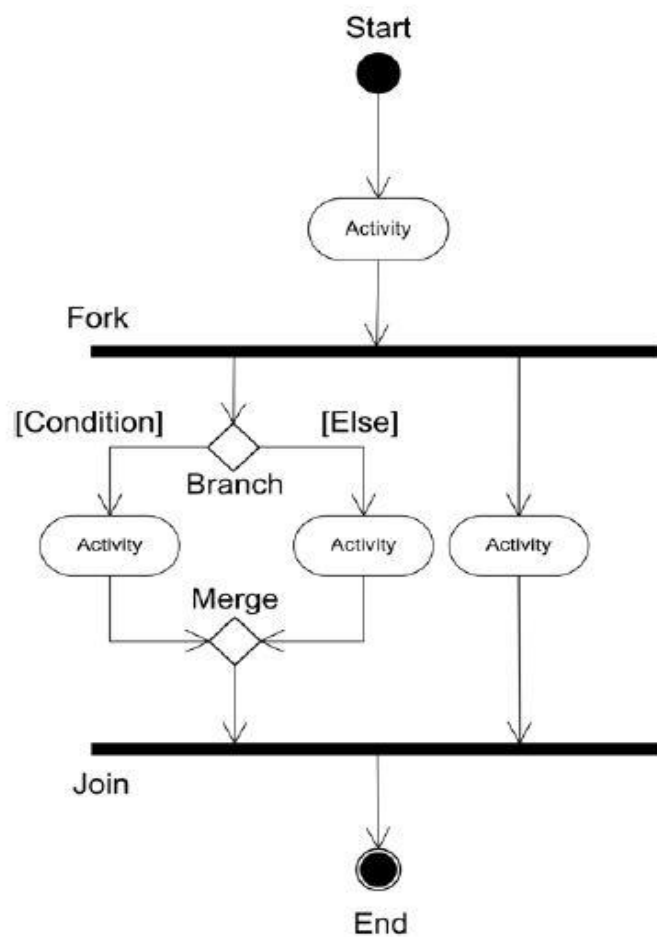
Pada contoh diatas *sequence diagram* dimulai ketika *student* melakukan *RequestToRegister* kemudian *Course Section* akan melakukan *create* pada *Registration*, lalu *Registration* akan menambah *Schedule* dan mengembalikan informasi *RegistrationList* kepada student.

2.14.4 Activity diagram

Teknik untuk menjelaskan *business process*, *procedural logic*, dan *work flow*. Bisa dipakai untuk menjelaskan teks use case dalam notasi grafis dengan menggunakan notasi yang mirip flow chart, meskipun terdapat sedikit perbedaan notasi (Hofmeister, Nord, & Soni, 1999).

- a. *Nodes*, menandakan *initial* dan *final node*, *final node* boleh lebih dari satu.
- b. *Activity*, aktivitas sistem dapat berupa aktivitas fisik juga bagi *user*.

- c. *Flow/edge*, arah sebuah proses.
- d. *Fork*, awal sebuah proses paralel.
- e. *Join* akhir proses paralel.
- f. *Condition*, kondisi yang dituliskan dalam bentuk teks
- g. *Decision*, implementasi *if* dan *then*.
- h. *Merge*, penyatuan beberapa *flow*.
- i. *Partition*, pemisahan antar *user* dan sistem yang menjalankan *activity*.



Gambar 2.16 Contoh Activity Diagram (Kendall & Kendall, 2005, p672)