

BAB II

LANDASAN TEORI

2.1 Barcode Dua Dimensi (2D)

Bercode 2 dimensi atau *Two Dimensional (2D)* disebut juga sebagai barcode matrix yang bisa dikatakan lebih canggih daripada barcode 1D, karena bisa memuat ratusan digit karakter yang disimpan dalam ruang yang kecil.

2.1.1 Quick Response (QR) Code

Menurut (soon,2008), kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR merupakan singkatan dari quick response atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, kode QR mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis kode QR dapat menampung informasi yang lebih banyak daripada kode batang.

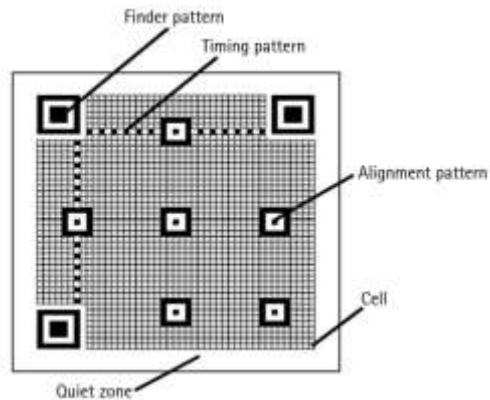
Teknik Labeling QR Code adalah teknik penamaan barang menggunakan label QR Code. (JSIKA, 2016) menjelaskan bahwa “Kode QR adalah pengembangan

kode bar yang biasanya terdiri atas garis-garis dengan ketebalan yang berbeda”. QR Code memiliki beberapa kelebihan dibandingkan barcode traditional, (Ariska, 2016) kelebihan tersebut antara lain:

- a. 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 symbol
- b. QR Code mampu menyandikan data hanya dengan membutuhkan sepersepuluh ruangan yang dibutuhkan oleh barcode biasa
- c. Mampu membaca pada arah manapun (360 derajat) QR Code mampu dibaca dalam berbagai arah secara cepat

2.1.1.1 Struktur QR Code

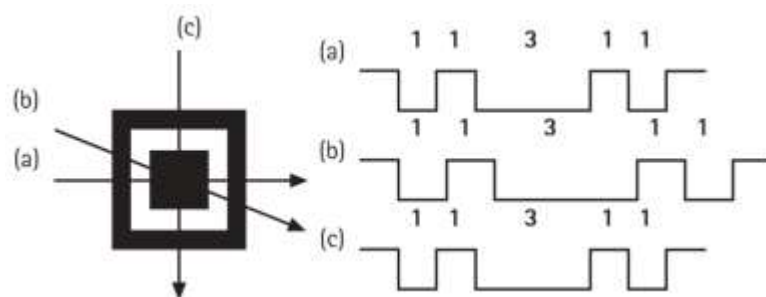
Menurut (soon, 2008), QR Code adalah simbol jenis matriks dengan struktur sel yang disusun dalam kotak. Ini terdiri dari pola fungsi untuk membuat pembacaan mudah dan area data tempat data disimpan. QR Code memiliki pola pencari, pola pelurusan, pola waktu, dan zona tenang.



Gambar 2.1 Struktur qr code (Soon,2008)

a. *Finder Pattern*

Pola untuk mendeteksi posisi QR Code. Dengan mengatur pola ini pada tiga sudut simbol, posisi, ukuran, dan sudut simbol dapat dideteksi. Pola finder ini terdiri dari struktur yang dapat dideteksi di semua arah (360°). Rasio antara hitam dan putih di antara garis pemindaian yang berjalan melalui pola finder selalu 1: 1: 3: 1: 1 bila dilihat dari arah mana saja di sekitar 360° yang mengelilinginya, dengan mendeteksi rasio khusus ini, pola penemu dapat dideteksi dari antara gambar yang ditangkap oleh sensor CCD (*area sensor*) untuk mengidentifikasi posisi Kode QR dalam waktu singkat.



Gambar 2.2 Finder Pattern (Soon,2008)

b. Alignment Pattern

Pola untuk memperbaiki distorsi QR Code. Ini sangat efektif untuk memperbaiki distorsi nonlinier. Koordinat pusat dari pola pelurusan akan diidentifikasi untuk memperbaiki distorsi symbol, dengan tujuan sel terisolasi hitam ditempatkan dalam pola pelurusan untuk membuatnya lebih mudah untuk mendeteksi koordinat pusat pola penyelarasan.

c. Timing Pattern

Suatu pola untuk mengidentifikasi koordinat pusat setiap sel dalam QR Code dengan pola-pola hitam dan putih yang disusun bergantian. Ini digunakan untuk memperbaiki koordinat pusat sel data ketika simbol terdistorsi atau ketika ada kesalahan untuk *pitch* sel. Ini diatur dalam arah vertikal dan horizontal.

d. Quiet Zone

Ruang margin yang diperlukan untuk membaca QR Code. Zona tenang ini membuatnya lebih mudah untuk memiliki simbol yang terdeteksi dari antara gambar yang dibaca oleh sensor CCD (*area sensor*). Empat atau lebih sel diperlukan untuk zona tenang.

e. Data Area

Data QR Code akan disimpan (*encoded*) ke dalam area data. Bagian abu-abu pada Gambar 11 mewakili area data. Data akan dikodekan ke dalam angka biner '0' dan '1' berdasarkan pada aturan encoding. Angka biner '0' dan '1' akan diubah menjadi sel hitam dan putih dan kemudian akan diatur. Area data akan memiliki kode Reed-Solomon yang digabungkan untuk data yang disimpan dan fungsi koreksi kesalahan.

2.1.2 Data Matrix Code

Data Matrix dikembangkan oleh perusahaan RVSI Acuity CiMatrix. Perusahaan ini kemudian diakuisisi oleh Siemens Energy and Automation, Inc. Data Matrix juga dikenal dengan nama ECC200 mengacu pada standar koreksi kesalahan yang diterapkan dalam spesifikasi Data Matrix terbaru. Sehubungan dengan pemasaran ponsel, simbologi Matriks Data juga diberi nama Semacode.

Data Matrix adalah simbologi barcode 2D dengan kepadatan data yang sangat tinggi. Simbol matriks data dicetak dalam bentuk persegi atau terkadang pola persegi panjang. Setiap titik dari simbol data matrix mewakili sedikit. Berbeda dengan barcode linier, dimana informasi dikodekan dalam rasio bar atau spasi satu sama lain. Biasanya titik hitam dalam simbol data matrix setara dengan *bit-value* 1, tetapi data matrix juga dapat dicetak putih pada hitam.

2.1.3 PDF417

PDF 417 dikembangkan tahun 1991 oleh perusahaan *Symbol*. Pada tahun 2007 perusahaan diakuisisi oleh Motorola. PDF417 juga dikenal dengan nama PDF-417, Portable Data File dan PDF 417. Istilah Macro PDF atau (Macro PDF417) digunakan ketika menggabungkan beberapa simbol barcode.

PDF417 adalah simbologi barcode 2D dengan kepadatan data sangat tinggi. Sebuah simbol PDF417 tunggal dapat dibayangkan sebagai beberapa kode batang linear disebut baris-baris yang ditumpuk di atas satu sama lain. Ini adalah alasan mengapa simbologi PDF417 kadang-kadang disebut (*stacked linear symbology*). Bentuk keseluruhan PDF417 berbentuk persegi panjang. Ukuran simbol dapat dimodifikasi oleh pengguna. Rasio lebar atau spasi satu sama lain menyandikan informasi dalam simbol PDF417. Akurasi pencetakan dan resolusi printer yang sesuai sangat penting untuk simbol PDF417 berkualitas tinggi. Simbol barcode 2-D murni seperti data matrix atau QR Code menyandikan informasi dengan adanya titik hitam atau putih pada posisi tertentu dan bukan dengan menggunakan bilah rasio ruang.

2.2 Inventarisasi

Menurut (Adhi, 2008), Inventarisasi merupakan proses mengelola pengadaan atau persediaan barang yang dimiliki oleh suatu kantor atau perusahaan dalam melakukan kegiatan operasionalnya. Tanpa adanya inventori suatu kegiatan usaha tidak akan terlaksana, untuk itu keberadaan inventori sangat penting. Inventaris barang adalah kegiatan melaksanakan pengurusan, penyelenggaraan, pengaturan,

pencatatan dan pendaftaran barang inventaris/hak milik, sedangkan daftar barang inventaris/hak milik adalah suatu dokumen berharga yang menunjukkan sejumlah barang milik organisasi dan dikuasai pimpinan organisasi yang berada di sub bagian-bagian, baik yang bergerak maupun yang tidak bergerak.

2.2.1. Tujuan Inventarisasi

Secara umum, inventarisasi dilakukan dalam rangka usaha penyempurnaan pengurusan dan pengawasan yang efektif terhadap sarana dan prasarana yang dimiliki oleh suatu sekolah. Secara khusus, inventarisasi dilakukan dengan tujuan-tujuan sebagai berikut:

- a. Menjaga dan menciptakan tertib administrasi sarana dan prasarana yang dimiliki oleh suatu sekolah.
- b. Menghemat keuangan sekolah baik dalam pengadaan maupun untuk pemeliharaan dan penghapusan sarana dan prasarana sekolah.
- c. Sebagai bahan atau pedoman untuk menghitung kekayaan suatu sekolah dalam bentuk materil yang dapat dinilai dengan uang.
- d. Memudahkan pengawasan dan pengendalian sarana dan prasarana yang dimiliki oleh suatu sekolah.

2.2.2 Manfaat Inventarisasi

Daftar inventarisasi barang yang disusun dalam suatu organisasi yang lengkap, teratur dan berkelanjutan dapat memberikan manfaat, yakni sebagai berikut:

- a. Menyediakan data dan informasi dalam rangka menentukan kebutuhan dan menyusun rencana kebutuhan barang.
- b. Memberikan data dan informasi untuk dijadikan bahan/pedoman dalam pengarahan pengadaan barang.
- c. Memberikan data dan informasi untuk dijadikan bahan/pedoman dalam penyaluran barang.
- d. Memberikan data dan informasi dalam menentukan keadaan barang (tua, rusak, lebih) sebagai dasar untuk menetapkan penghapusannya.
- e. Memberikan data dan informasi dalam rangka memudahkan pengawasan dan pengendalian barang.

2.3 Android

Menurut (hermawan,2011), android merupakan OS (*Operating System*) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

2.3.1 Arsitektur Android

Menurut (Nazruddin,2012), arsitektur yang ada pada *platform* android adalah sebagai berikut:

a. Application and Widgets

Application and widgets adalah layer dimana berhubungan dengan aplikasi dan biasanya download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasiditulis menggunakan bahasa pemrograman Java.

b. Application Frameworks

Applications Framework merupakan layer dimana para pembuat aplikasi menggunakan komponen-komponen yang ada di sini untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk di dalam Applications Framework adalah sebagai berikut:

- 1) *Views*
- 2) *Content Provider*
- 3) *Resource Manager*
- 4) *Notification Manager*
- 5) *Activity Manager*

c. Libraries

Libraries merupakan layer tempat fitur-fitur android berada. Pada umumnya libraries diakses untuk menjalankan aplikasi. Beberapa library yang terdapat pada android diantaranya adalah libraries Media untuk memutar media video atau audio, libraries untuk menjalankan tampilan, libraries Graphic, libraries SQLite untuk dukungan database, dan masih banyak library lainnya.

d. Android RunTime

Android RunTime merupakan layer yang membuat aplikasi android bisa dijalankan. Android RunTime dibagi menjadi dua bagian yaitu:

- 1) Core Libraries : berfungsi untuk menerjemahkan bahasa Java/C
- 2) Dalvik Virtual Machine : sebuah mesin virtual berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi pada Android secara efisien.

e. Linux kernel

Linux kernel adalah layer dimana inti dari operating sistem dari android itu sendiri, berisi file-file sistem yang mengatur sistem processing, memory, resources, drivers, dan sistem-sistem operanting andoroid lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6

2.4 Aplikasi

Menurut (Nugroho B, 2009), aplikasi berasal dari kata application yang artinya penerapan, lamaran, penggunaan, secara istilah aplikasi adalah program siap

pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

2.5 Basisdata

Menurut (Robi Yanto,2016), basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau *hard disk*).

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.5.1 Operasi Dasar Basis Data

Menurut (Fathansyah, 2002), didalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Didalam sebuah disk, kita dapat pula menempatkan beberapa (lebih dari satu) basisdata. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file/tabel. File/tabel inilah sesungguhnya data disimpan/ditempatkan, setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya, ada basis data kepegawaian, basis data akademik, basis data inventori (Pergudangan), dan sebagainya. Sementara dalam basis data akademik, misalnya, kita dapat menempatkan file mahasiswa, file mata_kuliah, file dosen, file jadwal, file kehadiran, file nilai, dan seterusnya.

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi:

- a. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru
- b. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada)
- c. Pembuatan file/tabel dari suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari sarsip yang telah ada
- d. Penghapusan file/tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip
- e. Penambahan/pengisian data baru ke sebuah file/tabel disebuah basis data (*insert*), yang identik dengna penambahan ke lemari arsip ke sebuah map arsip
- f. Pengambilan data dari sebuah file/tabel (*retrieve/search*) yang identik dengan pencarian lembaran arsip dari sebuah map arsip
- g. Pengubahan data dari sebuah file/tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip
- h. Penghapusan data dari sebuah file/tabel (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip

Operasi yang berkenaan dengan pembuatan objek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dank arena itu operasi-operasi inilah yang lebih

tepat mewakili aktivitas pengelolaan (*management*) dan pengolahan (*processing*) data dalam basis data.

2.5.2 Objektif Basis Data

Pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh menemukan kembali data (yang dicari) dengan mudah dan cepat. Disamping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan-tujuan lain.

Adapun pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) yang dikemukakan oleh (Robi Yanto, 2016), seperti berikut ini:

a. Kecepatan dan kemudahan (*speed*)

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika menyimpan data secara manual (non-elektronis) atau secara elektronik (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk spread sheet atau dokumen teks biasa).

b. Efisiensi ruang penyimpanan (*Space*)

Karena keterkaitan yang erat antara kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruangan penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Basis data efisiensi/optimalisasi

penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antara kelompok data yang saling berhubungan.

c. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constrain*) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

d. Ketersediaan (*Availability*)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu digunakan/butuhkan, maka dapat memilah adanya data utama/master/refereensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi gunakan, dapat diatur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi off-line) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan off-line (seperti removable disk, atau tape). Di sisi lain, karena kepentingan pemakaian data, sebuah geografis. Data nasabah sebuah bank, misalnya, dipisah-pisahkan dan disimpan di lokasi yang sesuai dengna keberadaan nasabah. Menggunakan dan

pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi/cabang, dapat juga di akses (menjadi tersedia/*available*) bagi lokasi/cabang lain.

e. Kelengkapan (*Completeness*)

Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relative (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama atau yang sekarang dianggap sudah lengkap belum tentu di masa yang akan datang juga demikian, pada sebuah basis data, disamping data harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek seperti struktur file/tabel atau indeks). Mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka tidak hanya dapat menambahkan record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

f. Keamanan (*Security*)

Sistem (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data, tetapi untuk sistem yang besar dan serius. Aspek keamanan juga dapat diterapkan dengan ketat, dengan begitu dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

g. Kebersamaan Pemakai (*Sharebility*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja, atau di satu lokasi saja atau boleh satu sistem/aplikasi saja, data pegawai dalam basis data kepegawaian, misalnya, dapat digunakan oleh banyak pemakai, dari sejumlah departemen dalam perusahaan atau oleh banyak sistem (sistem penggajian, sistem akuntansi, sistem inventori, dan sebagainya). Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi *deadlock* (karena banyak pemakai yang saling menunggu untuk menggunakan data).

2.6 *Web Service* atau *Web API*

Menurut (Hilmawan & Iskandar, 2014), *Web service* atau *Web API* adalah sebuah sistem perangkat lunak yang dapat didesain untuk mendukung interaksi yang terjadi pada sebuah organisasi yang akan dioperasikan melalui suatu jaringan. *Web Service* dapat diimplementasikan pada lingkungan internal untuk kebutuhan integrasi antar sistem aplikasi atau pada lingkungan eksternal. *Web service* digunakan oleh pengembang aplikasi untuk membantu aplikasi yang dibuat seperti memberikan *list* data ke aplikasi untuk diolah, melakukan pencarian suatu data.

Web service merupakan teknologi yang mengubah kemampuan *internet* dengan menambahkan kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola program-to-program (P2P). *Web service* memiliki protokol yang merupakan blok bangunan yang dapat melakukan komunikasi jarak jauh antara dua aplikasi yang ada :

- a. Layer 1: Extensible Markup Language (XML) protokol internet yang dapat membuat web service lebih mudah bertukar data antara aplikasi yang bervariasi dan dapat digunakan untuk menerjemahkan data tersebut.
- b. Layer 2: Simple Object Access Protocol (SOAP) berbasis XML dan digunakan untuk pertukaran data antara aplikasi jaringan.
- c. Layer 3: Web service Description Language (WSDL) digunakan untuk mendeskripsikan attribute layanan dan digunakan untuk menciptakan dokumen XML yang menggambarkan tugas yang dilakukan oleh web service.
- d. Layer 4: Universal Description, Discovery and Integration, yang mana merupakan digunakan untuk pembuatan direktori public atau privat.

2.7 JSON

JSON yang merupakan singkatan dari JavaScript Object Notation adalah standar berbasis teks untuk pertukaran data. Format JSON dikenal ringan (berukuran kecil), mudah untuk dibaca, ditulis, dan dipahami manusia serta mudah untuk diuraikan dan dibuat oleh mesin. Format ini dibuat berdasarkan bahasa pemrograman JavaScript, standar ECMA-262 edisi ketiga – Desember 1999. JSON bersifat

language independent namun menggunakan kaidah penulisan yang dikenal luas oleh programmer dari keluarga bahasa C (C, C++, C#, Java, JavaScript, Perl, Python, dll), hal tersebut menjadikan JSON sangat ideal sebagai bahasa dalam pengiriman data.

2.8 **Firestore Realtime Database**

Firestore Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, ketika membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.

2.9 **Konsep Perancangan Berorientasi Obyek**

2.9.1 *Unified Modelling Language (UML)*

Menurut (Hend, 2006), *Unified Modeling Language (UML)* adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak.

Menurut (Sholih, 2006) Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR.James Rumbough, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. UML menyediakan beberapa diagram yang menunjukkan berbagai aspek dalam system.

Unified Modeling Language (UML) adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke developer lainnya, tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML. Ada 5 (lima) macam diagram dalam *Unified Modeling Language* (UML) yaitu:

a. *Use Case Diagram*

Diagram ini memperlihatkan himpunan use case dan aktoraktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

b. *Class Diagram*

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi dan relasi-relasi antar objek.

c. *Sequence Diagram*

Diagram ini memperlihatkan interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

d. State Chart Diagram

Diagram ini memperlihatkan state-state pada sistem, memuat state, transisi, event, dan aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

e. Activity Diagram

Diagram ini memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

2.9.2 Use Case Diagram

Menurut (Munawar, 2005) Use case merupakan deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antar pengguna (yang disebut dengan actor) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkahlangkah yang menerangkan antar penggunaan dan sistem disebut skenario. Skenario mendeskripsikan kejadian, setiap urutan diinisiasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.



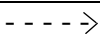

Menurut (Jacob et all, 1992), model use case adalah bagian dari model *requirement* termasuk disini adalah *problem domain object* model dan penjelasan






tentang *user interface*. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif *user*.

Use Case bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*, setiap *scenario* mendeskripsikan urutan kejadian, setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Secara singkat bisa dikatakan use case adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Simbol-simbol yang ada pada diagram use case dapat dilihat pada tabel 2.1 sebagai berikut:

Tabel 2.1 Simbol *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	Aktor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri
	<i>Generalization</i>	Hubungan dimana objek anak berperilaku dan struktur data dari objek yang adanya diatasya objek induk
	<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit
	<i>Extend</i>	Menspesifikasikan bahwa use case target

		memperluas perilaku dari use case sumber pada suatu titik yang diberikan
	<i>Association</i>	Apa yang menghubungkan objek satu dengan objek lain
	<i>System</i>	Menspesifikasikan paket yang menampilkan system secara terbatas
	<i>Use Case</i>	Dekripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Pembicaraan tentang use case, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

2.9.3 Sequence Diagram


Menurut (Sukamto dan Shalahuddin, 2013), “Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek”.

Sequence diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses


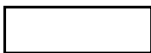

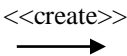
berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Simbol-simbol yang ada pada sequence diagram menurut Sukamto dan Shalahuddin (2013:165) dapat dilihat pada tabel 2.2 sebagai berikut:

Tabel 2.2 Simbol *Sequence Diagram*

SIMBOL	NAMA	KETERANGAN
	Aktor	Pengguna, 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

Tabel 2.2 Simbol *Sequence Diagram* (tabel lanjutan)

		merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
	<i>Lifeline</i>	Menyatakan kehidupan suatu objek
	Objek	Menyatakan objek yang berinteraksi pesan
	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
	<i>Message create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	<i>Message call</i>	Menyatakan suatu objek memanggil

1: nama_metode() →		operasi/metode yang ada pada objek lain atau dirinya sendiri
1 : masukan →	<i>Messege send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
1 : keluaran ----->	<i>Messege return</i>	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
<<destroy>> →	<i>Messege destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

2.9.4 Class Diagram

Menurut (Dennis, Alan; Wixom, Haley Barbara: & Tegarden, David, 2005) Class diagram merupakan suatu model statis yang menunjukkan class – class dan hubungan diantaranya dan senantiasa konstant di dalam sistem sepanjang waktu. Class diagram menggambarkan class berikut perilaku dan keadaan dengan menghubungkannya antar class – class.

Menurut Sukamto dan Shalahuddin (2013:141), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang

akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a. *Atribut* merupakan *variable* yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- a. Kelas main
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- b. Kelas yang menangani tampilan *system (view)*
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- c. Kelas yang diambil dari pendefinisian *use case (controller)*
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
- d. Kelas yang diambil dari pendefinisian data (*model*)
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Simbol-simbol yang ada pada *class diagram* menurut Rosa dan Shalahuddin (2013) dapat dilihat pada tabel 2.3:

Tabel 2.3 Simbol *Class Diagram*

SIMBOL	NAMA	KETERANGAN
<div style="border: 1px solid black; padding: 2px; width: fit-content;"> nama_kelas +attribut +operasi() </div>	Kelas	Kelas pada struktur sistem
○	Antarmuka	Sama dengan konsep interface dalam pemrograman berorientasi objek
—	Asosiasi	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
→	Asosiasi berarah	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
→▷	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
→→	Kebergantungan	Relasi antarkelas dengan makna kebergantungan antarkelas
→◇	Agregasi	Relasi antarkelas dengan makna semuabagian (whole-part)

2.10 Penelitian Terdahulu

Tabel 2.4 Penelitian Terdahulu

No	Nama Peneliti	Judul	Penjelasan
1.	Rizkysari Meimaharani dan Diana Laily	“E-Commerce Goody Bag Spunbond Menggunakan QR Code Berbasis Web Responsif”	Penelitian ini membuat aplikasi aplikasi <i>E-Commerce</i> yang dikembangkan merupakan sebuah kebutuhan yang selama ini diinginkan oleh Vantacy Shop untuk dapat membantu mempromosikan Goody Bag berbasis <i>Web responsive</i> yang mengimpleentasikan teknologi <i>QR Code</i> untuk mempermudah pihak Vantacy Shop dalam pendataan barang masuk dan barang keluar.

2.	Bambang Sugiantoro dan Fuad Hasan	“Pengembangan QR Code Scanner Berbasis Android Untuk Sistem Informasi Museum Sonobudoyo”	Penelitian ini membuat aplikasi QR Code Scanner yang dibangun berbasis Android menggunakan ZBar Library mempunyai fitur Scanner, Bookmark, tentang Museum Sonobudoyo, dan menampilkan foto koleksi secara 3D Rotate.
3.	Wina Sri Agustina, Rusda Wajhillah, Jamal Maulana Hudin	“Penerapan Teknik Labeling QR Code Berbasis Internet Pada Sistem Informasi Manajemen Aset RSUD. R. Syamsudin, SH. Sukabumi”	Penelitian ini membuat aplikasi sistem penamaan barang dan alat rumah sakit dengan menggunakan QR Code yang mampu memuat informasi barang inventaris secara lengkap. Pengurus barang hanya perlu menggunakan handphone berbasis Android untuk memindai QR Code pada barang inventaris.
4.	Agus Prihanto dan Dwi Fratianto	“Pengembangan Aplikasi Terpadu Untuk	Penelitian ini menjelaskan tentang aplikasi ruang baca yang dilengkapi fitur QR Code. Aplikasi terdiri dari 2

Tabel 2.4 Penelitian Terdahulu (tabel lanjutan satu)

		Meningkatkan Layanan Dan Akses Mahasiswa Terhadap Ruang Baca Dengan QR Code”	buah program utama yaitu, RAdmin digunakan oleh admin pengelola/pustakawan untuk transaksi simpan pinjam buku dan dapat mencetak kode QR untuk kartu anggota. Katalog digunakan oleh pengunjung untuk melakukan pencarian buku, melakukan absen dengan kartu anggota, penyerahan buku dan memberikan saran ke pengelola ruang baca
5.	Danniar Reza Firdausy, Satrio Agung Wicaksono	“Pengembangan Sistem Informasi Manajemen Pelaporan Sarana	Penelitian ni menghasilkan sistem informasi pelaporan dan pemeliharaan sarana prasarana yang berbasis aplikasi Android untuk pihak pelapor

	dan Fajar Pradana	dan Prasarana Studi pada Fakultas Ilmu Komputer Universitas Brawijaya”	dan <i>website</i> operator untuk pegawai. <i>website</i> operator bertujuan untuk memudahkan pegawai dalam menerima dan menindak lanjuti keluhan sarana prasarana yang dilaporkan.
6.	Thomas Muyumba dan Jackson Phiri	“A Web based Inventory Control System using Cloud Architecture and Barcode Technology for Zambia Air Force”	Penelitian ini menghasilkan aplikasi manajemen persediaan suku cadang dan merancang model <i>system</i> manajemen persediaan <i>barcode</i> otomatis. Inventaris berbasis web dikembangkan menggunakan model berbasis arsitektur cloud dan teknologi barcode untuk menyelesaikan masalah yang dihadapi oleh ZAF dalam manajemen inventaris
7.	Akhmad Qashlim dan Hasruddin	“Implementasi Teknologi QR-Code Untuk Kartu Identitas”	Penelitian ini menghasilkan kartu identitas yang dapat digunakan dalam mempermudah proses absensi mahasiswa, registrasi perpustakaan dan

Tabel 2.4 Penelitian Terdahulu (tabel lanjutan dua)

			mempermudah akses informasi proses belajar mengajar bagi para mahasiswa.
8.	Mukhamad Taqwa Nuddin dan Diana Laily Fithri	“Sistem Absensi Asistensi Dosen Menggunakan QR Code Scanner Berbasis Android Pada Program Studi Sistem Informasi Universitas Muria Kudus”	Penelitian ini tentang Sistem Absensi Asisten Dosen menggunakan QR Code <i>Scanner</i> untuk merekap jumlah kehadiran asisten dosen dan juga merekap honor yang di dapat berdasarkan kehadiran, Aplikasi ini menghasilkan informasi absen asisten dosen yang tepat karena tercantum waktu dan tidak bisa dirubah.
9.	Ardian Prima Atmaja dan	“Optimasi Aplikasi Simak-	Penelitian ini menghasilkan aplikasi SIMAK-BMN yang bersifat <i>stand</i>

	Fredy Susanto	BMN Untuk Inventarisasi Barang Milik Negara Berbasis Aplikasi Mobile Android ”	<i>alone</i> , yakni dengan mengembangkannya menjadi sistem monitoring BMN terpadu yang terdiri dari sistem berbasis web. Aplikasi <i>mobile</i> sebagai monitoring BMN yang dapat mencatat dan inventarisasi BMN menggunakan <i>smartphone</i> untuk membantu memudahkan pekerjaan dengan melakukan scanning QR Code dari tiap-tiap BMN.
10.	Arief Budiman dan Joko Triono	“Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android”	Penelitian ini menghasilkan aplikasi berbasis <i>mobile</i> yang dapat dioperasikan pada <i>smartphone</i> berbasis android, sehingga manajemen parkir di lingkungan Universitas Merdeka Madiun dapat lebih efektif dan efisien. Petugas parkir dapat mengontrol kendaraan masuk dan keluar sehingga dapat memperkecil kemungkinan terjadinya tindak pencurian.

2.11 State Of The Art

Tabel 2.5 terdapat jurnal penelitian terdahulu yang berkaitan dengan *Quick Response (QR) Code* dan penelitian ini dengan memberikan penjelasan untuk mengetahui perbedaan penelitian diantara keduanya.

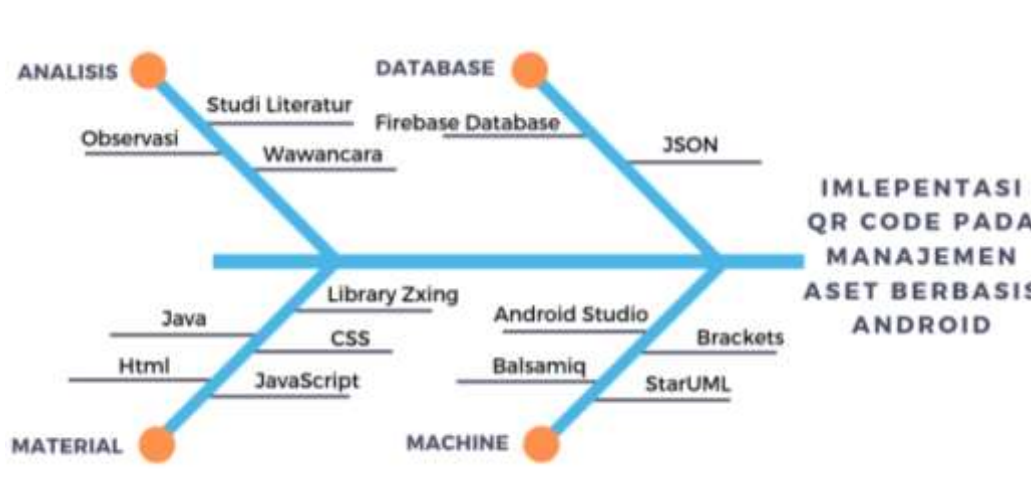
Tabel 2.5 State Of The Art

No	Peneliti	Judul	Penjelasan
1	Bambang Sugiantoro dan Fuad	“Pengembangan QR Code Scanner Berbasis Android	Penelitian ini membuat aplikasi QR Code Scanner yang dibangun berbasis Android menggunakan ZBar Library mempunyai

	Hasan	Untuk Sistem Informasi Museum Sonobudoyo”	fitur Scanner, Bookmark, tentang Museum Sonobudoyo, dan menampilkan foto koleksi secara 3D Rotate.
2	Farid Saeful Gani, Aldy Putra Aldya, Rahmi Nur Shofa	“Implementasi <i>QR Code</i> Pada Aplikasi Inventarisasi Barang Berbasis Android (Studi Kasus SMA Negeri 1 Banjarsari)”	Penelitian ini membuat aplikasi inventarisasi barang yang dapat membantu dalam pendataan barang dengan mengimplementasikan <i>QR Code</i> pada aplikasi android untuk dapat menampilkan informasi barang dan <i>update</i> kondisi barang. aplikasi ini menggunakan database <i>firebase real-time database</i> sebagai penyimpanan datanya.

2.12 Diagram Fishbone

Berikut ini diagram fishbone Implementasi *Quick Response (QR) Code* pada aplikasi inventarisasi barang berbasis android dapat dilihat pada gambar 2.3.



Gambar 2.3 Diagram Fishbone

Gambar 2.3 menunjukan diagram tulang ikan (*fishbone*) penelitian, diagram ini menggambarkan identifikasi berbagai permasalahan dan langkah penyelesaiannya untuk mencapai tujuan penelitian.