

BAB 2

LANDASAN TEORI

2.1 *Commuterline*

PT KAI Commuter Jabodetabek adalah salah satu anak perusahaan di lingkungan PT Kereta Api Indonesia (Persero) yang mengelola KA Commuter Jabodetabek. KCJ dibentuk sesuai dengan Inpres No. 5 tahun 2008 dan Surat Menteri Negara BUMN No. S-653/MBU/2008 tanggal 12 Agustus 2008.

Pembentukan anak perusahaan ini berawal dari keinginan para stakeholdernya untuk lebih fokus dalam memberikan pelayanan yang berkualitas dan menjadi bagian dari solusi masalah transportasi perkotaan yang semakin kompleks. Perseroan ini resmi menjadi anak perusahaan PT KERETA API (Persero) sejak tanggal 15 September 2008.

Kehadiran KCJ dalam industri jasa angkutan KA Commuter bukanlah kehadiran yang tiba-tiba, tetapi merupakan proses pemikiran dan persiapan yang cukup panjang. Dimulai dengan pembentukan Divisi Angkutan Perkotaan Jabotabek oleh PT KAI (Persero), yang terpisah dari PT KAI (Persero) Daop 1 Jakarta.

Setelah pemisahan ini, pelayanan KRL di wilayah Jabotabek berada di bawah PT KAI (Persero) Divisi Angkutan Perkotaan Jabotabek sementara pelayanan KA jarak jauh yang beroperasi di wilayah Jabodetabek berada di bawah PT KAI Daop 1 Jakarta.

Dan akhirnya PT KAI (Persero) Divisi Angkutan Perkotaan Jabotabek berubah menjadi sebuah perseroan terbatas, PT KCJ. Setelah menjadi perseroan terbatas, perusahaan ini mendapatkan izin usaha No. KP 51 Tahun 2009 dan izin

operasi penyelenggara sarana perkeretaapian No. KP 53 Tahun 2009 yang semuanya dikeluarkan oleh Menteri Perhubungan Republik Indonesia.

Tugas pokok perusahaan yang baru ini adalah menyelenggarakan perusahaan pelayanan jasa angkutan kereta api komuter dengan menggunakan sarana Kereta Rel Listrik di wilayah Jakarta, Bogor, Depok, Tangerang, dan Bekasi (Jabodetabek) serta perusahaan di bidang usaha non angkutan penumpang.

KCJ Memulai modernisasi angkutan KRL pada tahun 2011 dengan menyederhanakan rute yang ada menjadi lima rute utama, penghapusan KRL ekspres, penerapan kereta khusus wanita, dan mengubah nama KRL ekonomi-AC menjadi kereta Commuter Line. Proyek ini dilanjutkan dengan renovasi, penataan ulang, dan sterilisasi sarana dan prasarana termasuk jalur kereta dan stasiun kereta yang dilakukan bersama PT KAI (persero) dan Pemerintah.

Pada 1 Juli 2013. KCJ mulai menerapkan sistem tiket elektronik (E-Ticketing) dan sistem tarif progresif. Penerapan dua kebijakan ini menjadi tahap selanjutnya dalam modernisasi KRL Jabodetabek.

Hingga Oktober 2016, KCJ telah memiliki 826 unit KRL, dan akan terus bertambah. Sepanjang tahun 2016, KCJ telah melakukan penambahan armada sebanyak 60 kereta. Hal ini untuk memenuhi permintaan penumpang yang terus bertambah dari waktu ke waktu.

Pada tahun 2016, rata-rata jumlah pengguna KRL per hari mencapai 850.000 pengguna pada hari-hari kerja, dengan rekor jumlah pengguna terbanyak yang dilayani dalam satu hari adalah 931.082. Sebagai operator sarana, kereta Commuter Line yang dioperasikan KCJ saat ini melayani 72 stasiun di seluruh Jabodetabek dengan jangkauan rute mencapai 184,5 km.

Dengan mengusung semangat dan semboyan *Best Choice for Urban Transport*, KCJ saat ini terus bekerja keras untuk memenuhi target melayani 1,2 juta penumpang per hari pada tahun 2019[3].

2.2 Java



Gambar 2.1 logo Pemrograman Java

Sebagian besar bahasa pemrograman, sebuah kode program harus dikompilasi (*compile*) maupun diterjemahkan (*interpret*) sehingga dapat dijalankan di dalam sebuah komputer. Software yang digunakan untuk kompilasi pada kode program disebut *compiler*, sedangkan yang digunakan untuk melakukan interpretasi disebut dengan *interpreter*. Perbedaan antara *interpreter* dan *compiler* adalah terletak pada proses eksekusi kodenya.

Java menerapkan dua proses tersebut ini berarti bahwa Java berperan sebagai *compiler* dan juga sebagai *interpreter*. Java menggunakan konsep yang sederhana, pertama kode program ditulis dengan bahasa Java (berekstensi *.java*) akan dikompilasi oleh *compiler* menjadi suatu kode objek. Kode objek ini disebut dengan istilah *bytecode* yang hasil akhir programnya berekstensi *.class* bukan file *.exe*.

Dengan adanya konsep *bytecode* ini, dalam *terminology* Java dikenal adanya istilah “*write once run anywhere*”. Ini berarti bahwa sekali kita menulis program Java dan melakukan kompilasi terhadapnya maka *bytecode*-nya dapat dijalankan di dalam *platform* manapun selama *platform* tersebut memiliki JVM (*Java Virtual Machine*) [15].

Hingga saat ini bahasa pemrograman java terus mengalami peningkatan. Mula-mula diproduksi Java 1, yaitu yang biasa disebut JDK. Pada perkembangan selanjutnya *Sun Microsystem* memperkenalkan Java versi 1.2 atau yang lebih dikenal dengan java 2 yang terdiri atas JDK dan JRE. Pada Java 2 ini dibagi menjadi 3 kategori yaitu:

1. *Java Standard edition (J2SE).*

Kategori Java ini digunakan untuk menjalankan dan mengembangkan aplikasi Java pada level PC (*Personal computer*).

2. *Java Enterprise Edition (J2EE).*

Kategori Java ini digunakan untuk menjalankan dan mengembangkan program aplikasi Java pada lingkungan *enterprise* atau perusahaan.

3. *Java 2 micro Edition (J2ME).*

Kategori ini digunakan untuk menjalankan dan mengembangkan program aplikasi java pada perangkat-perangkat semacam ponsel, PDA, dan Pocket PC.

2.2.1 JRE (*Java Runtime Environment*)

JRE (*Java Runtime Environment*) adalah sesuatu yang memungkinkan sebuah program Java dapat berjalan di mesin Anda. JRE ini mengeksekusi *binary-binary* dari *class-class* dan mengirimnya ke *Java Virtual Machine*(JVM) agar diproses lagi ke prosesor. Setiap JRE pasti memiliki sebuah JVM di dalamnya untuk melakukan pemrosesan selanjutnya.

2.2.2 JDK (*Java Development Kit*)

Seperti namanya, JDK adalah semacam *library (kit)* yang digunakan dan dibutuhkan untuk *development*. JDK ini berguna saat menulis kode program. Seperti halnya JRE, JDK juga memiliki JVM di dalamnya.

Pada Android, penggunaan bahasa pemrograman java tidak berbeda dengan java untuk *platform* lainnya, yaitu membutuhkan dua buah produk yang dikeluarkan untuk membantu dalam membuat program aplikasi[15].

2.3 Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang dikeluarkan oleh Google Inc pada bulan November 2007, bersifat *open source* sehingga para pengembang dapat membuat program aplikasi sendiri untuk perangkat *mobile* sesuai dengan kebutuhan[14]. Pemrograman perangkat ini menggunakan bahasa pemrograman *Extensible Markup Language*(XML) dan *Java* yang dikhususkan untuk *platform* ini, sehingga program aplikasi yang dibuat tidak dapat dijalankan pada perangkat *mobile* lain seperti: *Symbian OS* dan *Blackberry OS* yang juga mendukung program aplikasi berbasis *Java*. Terdapat dua distributor sistem operasi ini, yaitu: *GMS*(*Google Mail Services*) yang mendapat dukungan langsung dari Google dan *OHD*(*Open Handset Distribution*) yang tidak mendapat dukungan langsung dari Google dan melakukan pendistribusian sistem operasi ini secara bebas[14].

2.3.1 Keunggulan Android

Android dipuji sebagai platform *mobile* pertama yang lengkap, terbuka, dan bebas.

1. Lengkap (*Complete Platform*)

Para *designer* dapat melakukan pendekatan yang komperhensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan program aplikasi.

2. Terbuka (*Open Source Platform*)

Platform Android disediakan melalui lisensi *open source*. Pengembangan dapat dengan bebas untuk mengembangkan program aplikasi Android sendiri menggunakan Linux Kernel 2.6

3. Bebas (*Free Platform*)

Android adalah *platform* atau program aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform*

Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Program aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.4 Versi Android

Versi pertama yaitu Android v1.0 *Astro (Alpha)*. Kedua adalah Android v1.1 *Bender (Beta)*. Versi ini dirilis pada tanggal 5 November 2007 yang merupakan versi lanjutan dari Android *Astro (Alpha)*. Ketiga adalah Android v1.5 *Cupcake* yang dirilis pada tanggal 30 April 2009. Keempat adalah Android v1.6 *Donut* yang dirilis pada tanggal 15 September 2009. Kelima adalah Android v2.0 – 2.1 *Eclair* yang dirilis pada tanggal 26 Oktober 2009. Ke enam adalah Android v2.2 *Frozen Yoghurt (Froyo)* yang dirilis pada tanggal 20 Mei 2010. Ke tujuh adalah Android v2.3 *Gingerbread*. Dirilis pada tanggal 6 Desember 2010. Ke delapan adalah Android v3.0 – 3.2 *Honeycomb*. Ke sembilan adalah Android v4.0 *Ice Cream Sandwich*. Dirilis pada 16 Desember 2011. Ke sepuluh adalah Android v4.1 – 4.3 *Jelly Bean* Dirilis pada 9 Juli 2012. Ke sebelas adalah Android v4.4 *Kitkat* yang dirilis pada tanggal 31 Oktober 2013. Ke dua belas adalah Android v5.0 – 5.1 *Lollipop*. Dirilis pada tanggal 15 Oktober 2014. Ke tiga belas adalah Android v6.0 *Marshmallow*[5].

2.5 Android SDK

Android SDK adalah *tools API(Application Programming Interface)* yang diperlukan untuk memulai mengembangkan program aplikasi pada *platform*. Android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan program aplikasi kunci yang rilis oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan program aplikasi pada platform Android menggunakan bahasa pemrograman java. Sebagai *platform* program aplikasi-netral, Android memberi anda kesempatan untuk membuat program aplikasi yang kita butuhkan yang bukan merupakan program aplikasi bawaan *handphone* atau *smartphone*[14].

Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk *desktop* modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks *editor* untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* program aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh)[14].

2.6 Android Studio IDE

Android studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform* Android yang dikembangkan dari IDE IntelliJ IDEA. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android Studio bersifat free dibawah Apache License 2.0. Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Android Studio menawarkan beberapa fitur dan kelebihan seperti berikut:

1. Dikembangkan IntelliJ IDEA Community Edition, Java IDE yang populer oleh JetBrains.
2. Fleksibel *build system* berbasis Gradle.
3. Membuat beberapa APK dalam *project* yang sama.
4. Editor kode cerdas yang mampu menyelesaikan kode, *refactoring*, dan analisis kode.
5. Diperluas dukungan template untuk layanan Google dan berbasis jenis perangkat seperti ponsel Android, tablet, Android Wear, Android TV, Android Auto dan Google Glass.
6. Terintegrasi dengan GitHub. Kita bisa memulai proyek menggunakan

kode template bahkan mengimpor contoh kode Google dari gitHub.

7. Layout editor kaya dengan dukungan ntuk mengedit tema.
8. Built-in mendukung untuk *Google Coud Platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*.

2.7 XML

XML (*eXtensible Markup Language*) dikembangkan mulai dari tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998[9]. XML merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis sebagai proyek berskala besar. Pada saat HTML dikembangkan pada tahun 1990, para peneliti XML menggunakan bagian paling penting pada SGML dan dengan berdasarkan pada pengembangan HTML yang menghasilkan *markup language* yang tidak kalah baik dengan SGML. Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan *tag* pembuka dan *tag* penutup serta atribut elemen. Perbedaan nya adalah HTML mendefinisikan dari awal *tag* dan atribut yang dipakai di dalamnya, sedangkan pada XML kita bisa menggunakan *tag* dan atribut sesuai kehendak kita..

2.7.1 Keunggulan XML

XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalu HTML digunakan untuk menampilkan informasi dan berfokus pada bagian informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar layaknya *heading*, paragraf, tabel, dan lain sebagainya.

Sama dengan HTML, *file* XML berbentuk teks sehingga bila diperlukan kita bisa membacanya tanpa memerlukan bantuan *software* khusus. Hal ini memudahkan pengembang program aplikasi menggunakan XML untuk men-*debug* programnya. XML lebih fleksibel dibanding HTML dalam hal kemampuannya

menyimpan informasi data. Pada XML kita bisa menyimpan data baik dalam atribut maupun sebagai isi elemen yang diletakkan diantara *tag* pembuka dan *tag* penutup. Tag yang tersedia di Android dan sering digunakan antara lain ada pada tabel 2.1.

Kelebihan lain yang dimiliki XML adalah bahwa informasi bisa di pertukarkan dari satu sistem ke sistem lain yang berbeda *platform*. Misalnya dari Windows ke Unix, atau dari PC ke Macintosh bahkan dari internet ke ponsel dengan teknologi WAP[9].

Selain nama-nama tag, pada XML Android juga menyediakan perintah untuk mengatur *widget-widget* yang dituliskan di dalam *tag-tag* tersebut. Perintah-perintah yang sering digunakan antara lain ada pada tabel 2.2.

No.	Perintah	Fungsi
1.	<i><LinearLayout</i>	jika jenis ini yang digunakan, <i>widget</i> akan terurut secara vertikal berdasarkan urutan penulisan <i>widget</i>
2.	<i><RelativeLayout</i>	tata letak <i>widget</i> relatif, bergantung <i>widget</i> lainnya
3.	<i><TextView</i>	membuat <i>widget</i> teks
4.	<i><ImageView</i>	membuat <i>widget</i> untuk menampilkan gambar

Tabel 2.1 *Tag* XML pada Android

No.	Perintah	Fungsi
1.	<i>android:id</i>	memberikan nama <i>id</i>
2.	<i>android:height</i>	mengatur tinggi <i>widget</i>
3.	<i>android:layout_weight</i>	mengatur lebar <i>widget</i>
4.	<i>android:text</i>	memberikan teks
5.	<i>android:textSize</i>	mengatur ukuran teks
6.	<i>android:textStyle</i>	mengatur gaya teks
7.	<i>android:textColor</i>	mengatur warna teks
8.	<i>android:gravity</i>	mengatur posisi <i>widget</i> secara vertikal
9.	<i>android:layout_gravity</i>	mengatur posisi <i>widget</i> terhadap layar
10.	<i>android:layout_width</i>	mengatur lebar layar
11.	<i>android:layout_height</i>	mengatur tinggi layar
12.	<i>android:background</i>	memberi <i>background</i> pada layar

Tabel 2.2 Perintah XML untuk Mengatur *Widget*

2.8 Diagram UML

pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk mengspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung[16].

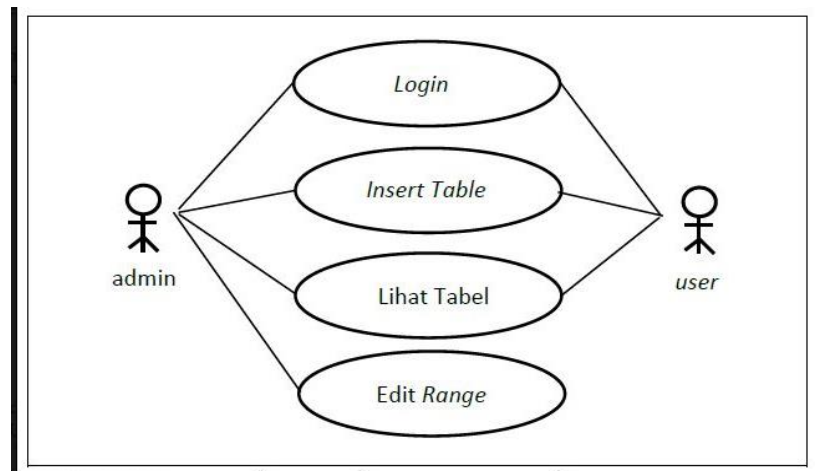
UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek [16].

Seperti yang kita ketahui bahwa banyak hal di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan lingkungan, lingkungan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualisasikan. Secara analogi jika dengan bahasa yang kita gunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalah hal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi di dalam bidang sistem informasi merupakan hal yang sangat wajar [16].

2.8.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya [16].

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Contoh dari Use Case Diagram:



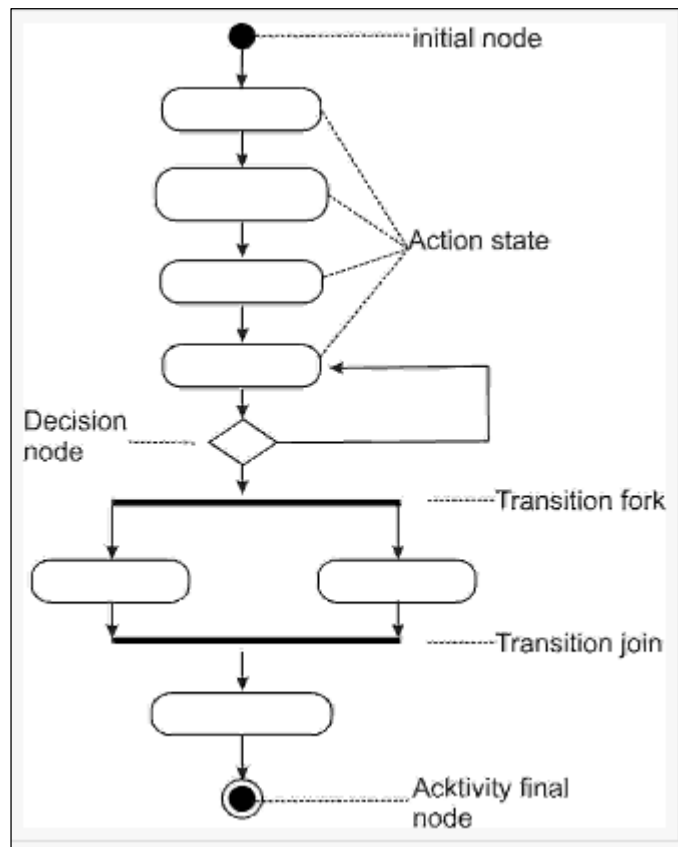
Gambar 2.2 contoh Use Case Diagram

2.8.2 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[16].

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Contoh dari Activity Diagram:



Gambar 2.3 contoh *Activity Diagram*

2.9 Corel Draw



Gambar 2.4 logo aplikasi grafis CorelDraw

CorelDRAW – The Popular Drawing Software, CorelDRAW adalah sebuah drawing software yang sangat populer. Sebuah software yang serba guna dan dimanfaatkan oleh para desainer dari berbagai aliran: senimurni, desain logo, desain ikon dan karakter, desain poster, brosur, kartu nama, cover buku dan sebagainya.

Digemari oleh desainer cetak karena memiliki paletwarna yang mendekati warna yang dihasilkan oleh mesin cetak, meskipun demikian CorelDRAW tidaklah cocok digunakan untuk layout isi buku. Pertama kali muncul pada tahun 1987, versi terakhir yang dirilis adalah X7 (17). Popularitas dan penggunaannya yang begitu luas membuat software lain yang sejenis, sedikit banyak kemudian mengikuti tampilan dan cara penggunaannya, termasuk drawing software yang tidak berbayar (open source) [4].

2.10 Google Play Services

Google Play Services merupakan API. Layanan Google Play digunakan untuk memperbaharui aplikasi google dan aplikasi dari google play.

Komponen ini menyediakan fungsi inti seperti otentikasi Google Services, sinkronisasi kontak, akses ke semua pengaturan pribadi pengguna terbaru, dan juga layanan berbasis lokasi dengan menggunakan sumber daya yang minim[7].

2.11 API (*Application Programming Interface*)

Antarmuka pemrograman aplikasi (Application Programming Interface/API) adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk system operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi.

API juga dapat diartikan sebagai sekumpulan perintah, fungsi, class dan protocol yang memungkinkan suatu software dapat menggunakan fasilitas dan saling berhubungan dengan software lainnya. Tujuan dari API adalah untuk menghilangkan kerumitan dari sistem dengan cara membuat sebuah blok besar yang terdiri dari software di seluruh dunia dan menggunakan kembali perintah, fungsi, class atau protokol yang API miliki, dengan cara ini programmer tidak perlu lagi membuang waktu untuk membuat dan menulis infrastuktur sehingga akan menghemat waktu kerja dan lebih efisien[11].

2.12 Google API

Google api adalah sekumpulan aplikasi antarmuka pemrograman (APIs) dikembangkan oleh Google yang memungkinkan komunikasi dengan layanan Google dan integrasi mereka ke layanan lain. Contoh pencarian, Gmail, menerjemahkan atau Google Maps. Aplikasi pihak ketiga dapat menggunakan api untuk mengambil keuntungan dari atau memperluas fungsi layanan yang ada[13].

Api menyediakan fungsi seperti analytics, mesin belajar sebagai layanan (prediksi API) atau akses ke data pengguna (ketika diberikan izin untuk membaca data). Contoh penting lain adalah Google peta tertanam di website, yang dapat dicapai dengan menggunakan API maps statis, tempat API atau Google Earth API.

2.13 Geofencing

Geofencing sejak tahun 2002, *geofencing* bisa diartikan sebagai geo-pagar yang bekerja dengan gambar virtual pagar di sekitar area tertentu menggunakan GPS, melayani pesan hanya keperangkat yang berada dalam ruang ini geografis. Geofence pagar atau batas-batasnya dapat dibuat secara dinamis , seperti dalam radius sekitar sebuah posisi atau titik lokasi tertentu. *Geofencing* ini juga menggunakan layanan berbasis lokasi (LBS) atau *Location Base Services* yang dapat menentukan kita masuk atau keluar dari area *geofence*[5].

Dengan fitur *Geofencing* dapat membuat perimeter virtual pada sebuah area geografis yang anda inginkan pada peta. Tujuan utama untuk membuat *geofencing* adalah untuk memonitor keberadaan kendaraan atau seseorang apakah dalam area yang seharusnya atau tidak. Ketika kendaraan atau seseorang keluar atau masuk kedalam area *geofencing* maka system akan membuat dan mengirim peringatan baik berupa pesan atau alarm[5].

Geofence ini memiliki 3 fitur, yang pertama adalah *enter* yang berfungsi untuk memberikan notifikasi apabila sudah memasuki daerah atau radius yang ditentukan. Yang kedua adalah *dwell* yang berfungsi memberikan notifikasi ketika beberapa saat didalam daerah atau radius. Yang ketiga adalah *exit* digunakan untuk memberikan notifikias pada saat keluar dari daerah atau radius dari *geofence*.

2.14 *Global Positioning System (GPS)*

GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaanya dimana lokasinya berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun pengguna tersebut berada, maka *GPS* bisa membantu menunjukkan arah. Layanan *GPS* ini tersedia gratis.

Awalnya *GPS* hanya digunakan hanya untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. *GPS* dapat digunakan dimanapun juga dalam 24 jam. Posisi unit *GPS* akan ditentukan berdasarkan titik-titik koordinat *latitude* dan *longitude*[12].

2.15 *Location Based Services (LBS)*

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning services* (GPS) dan *cell-based* location dari Google[10].

Selain itu, LBS tersebut terdiri dari beberapa komponen diantaranya *mobile devices*, *communicatin network*, *position component* dan *services and content provider*. *Mobile device* merupakan komponen yang sangat penting, piranti mobile tersebut diantaranya adalah smartphone dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning service* (GPS) dan *cell-based location* dari Google.

Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan *service and content*

provider. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya.

Sederhananya, dengan layanan LBS kita dapat mengetahui posisi dimana kita berada, posisi teman, dan posisi rumah sakit atau pom bensin yang jaraknya dekat dengan kita. Dalam mengukur posisi, digunakan lintang dan bujur untuk menentukan lokasi geografis. Tetapi, Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, kita dapat mengkonversi nilai lintang bujur menjadi alamat dunia nyata atau sebaliknya[10].

2.16 Alarm

Menurut KBBI(Kamus Besar Bahasa Indonesia) alarm adalah tanda bahaya berupa sinyal, bunyi, sinar. Alat mekanik yg dirancang untuk memperingatkan akan adanya bahaya atau kerusakan. Sedangkan bel alarm adalah bel yang memberikan tanda peringatan berupa bunyi atau sinar[1].

Kelebihan Alarm Dapat memberikan peringatan dini terhadap bahaya yang akan terjadi sehingga manusia dapat mengantisipasi dan meminimalisir korban jiwa maupun kerugian harta benda. Kelemahan Alarm Alarm merupakan alat yang mampu menyebabkan reaksi positif dan negatif pada manusia. Orang yang mendengar bunyi alarm yang nyaring dapat mengeluarkan reaksi panik dan menyelamatkan diri secara tidak rasional yang dapat membahayakan dirinya Jenis – jenis alarm adalah sebagai berikut[8]:

1. Alarm rumah

Jenis alarm ini biasanya ditempel di area rumah untuk menjamin dan melindungi properti yang ada di dalam rumah dari bahaya pencurian. Sistem alarm rumah dapat memonitor keamanan rumah dan memberikan laporan kepada pemilik jika terjadi tindakan yang mengancam keamanan rumah sehingga hal-hal yang tidak diinginkan dapat diminimalisir. Cara kerja alarm rumah terbagi dua[8], yaitu:

- Sensor infra merah Sistem pendeteksi bahaya yang digunakan adalah infra merah yang dalam bahasa teknis disebut PIR (Passive Infra Red) – infra merah pasif. Sensor ini hanya untuk mendeteksi saja sehingga disebut pasif. Sensor infra merah memiliki keuntungan dari sisi penguasaan area di mana sensor ini mempunyai jarak radius 15m/90 derajat dan biasanya dipasang di tempat strategis.

- Sistem alarm GSM Jenis alarm tanpa kabel yang dalam instalasinya relatif lebih mudah. Kelebihan alarm ini, ketika terjadi bahaya, maka alarm akan memberikan laporan kepada pemilik melalui jaringan GSM selama pemilik berada dalam jangkauan sinyal GSM.

2. Alarm mobil

Alarm mobil di Indonesia baru dikenal pada pertengahan 1980-an. Jenis alarm terus berevolusi sejak tahun 90-an di Amerika mulai dari hanya bermodalkan sebuah tombol yang dihubungkan dengan jaringan listrik mobil, alarm dengan pengendali jarak jauh berupa remote, serta fasilitas alat pengaman yang aktif otomatis dalam 30 detik setelah pintu mobil ditutup. Fungsi alarm mobil adalah untuk melindungi harta benda yang ada di dalam mobil dan mencegah pencurian mobil[8].

3. Alarm kebakaran

Sistem pengindra api (bahasa inggris – fire alarm system) merupakan sebuah sistem terintegrasi yang didesain untuk mendeteksi adanya gejala kebakaran. Alarm tersebut memberikan peringatan dalam sistem evakuasi dan dilanjutkan dengan sistem instalasi pemadam kebakaran secara otomatis maupun

manual (bahasa inggris – fire fighting system). Peralatan utama dalam pengendali sistem ini adalah Main Control Fire Alarm (MCFA) atau Fire Alarm Control Panel (FACP) yang menerima sinyal masuk (input) dari semua detektor serta komponen pendeteksi dan kemudian memberikan sinyal keluar (output) melalui komponen keluaran yang sudah ditetapkan[8].

4. Alarm banjir

Kegunaannya adalah untuk memperingatkan apabila akan terjadi banjir. Cara kerjanya menggunakan sensor air hujan dan akan memberi sinyal apabila debit air hujan yang turun sudah melebihi batas[8].

5. Alarm gempa

Alarm ini berfungsi untuk memberikan peringatan adanya getaran (gempa) bumi. Cara kerjanya yaitu dengan mendeteksi kedatangan gelombang seismik P-Wave sebelum kedatangan S-Wave dan Surface Wave (Q-Wave dan R-Wave) di mana gelombang tersebut bersifat berbahaya dan merusak. Sistem alarm dengan speaker otomatis yang bersuara sangat nyaring mampu membangunkan orang yang sedang tidur sehingga dapat segera menyelamatkan diri[8].

6. Alarm bayi

Sistem pemancar yang terdapat pada alarm bayi digunakan untuk mendengarkan suara yang ditimbulkan oleh bayi dari jarak jauh. Pemancar ini dikendalikan dari jarak jauh dan mampu memantau kondisi dan keberadaan bayi sehingga orang tua dapat mengetahui kondisi bayinya ketika menangis ataupun tertidur[8].

7. Alarm komputer

Program alarm jam umumnya sudah terdapat pada komputer pribadi. Komputer dapat berfungsi sebagai alarm secara digital untuk memberi peringatan akan sesuatu dengan berbagai macam bentuk dan bunyi yang dapat disesuaikan[8].

8. Alarm online

Program alarm dalam jaringan atau online yang bisa diunduh melalui jaringan internet. Dapat dipasang pada komputer pribadi yang terhubung pada jaringan internet[8].

9. Alarm telepon genggam

Telepon genggam modern memiliki fitur alarm yang dapat dimatikan secara manual. Bunyi alarm dapat disesuaikan dengan keinginan pemilik telepon genggam. Kegunaan alarm pada telepon genggam adalah untuk mengingatkan pemilik akan suatu jadwal ataupun memperingatkan pada jam-jam tertentu seperti alarm jam untuk membangunkan tidur[8].

10. Alarm jam

Alarm ini disebut juga alarm clock (bahasa inggris) yaitu sebuah jam yang didesain untuk mengeluarkan suara nyaring pada jam-jam tertentu. Gunanya untuk membangunkan orang dari tidurnya di pagi hari. Alarm jam dapat juga digunakan untuk membangunkan orang dari tidur siang dan juga sebagai tanda pengingat sebuah jadwal. Untuk menghentikan bunyi alarm, orang harus menekan tombol di badan jam dan alarm akan mati beberapa waktu setelahnya[8].