BAB 2
TINJAUAN PUSTAKA

2.1 Profil Rumah Sakit Umum Daerah Gunung Jati

2.1.1 Sejarah Instansi

Pembangunan Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon, awalnya diajukan oleh Dewan Kota pada tahun 1919 dan kemudian pada tanggal 14 Maret 1920 dilaksanakan peletakan batu pertama pembangunan gedung rumah sakit yang terletak di Jalan Kesambi. Rumah sakit selesai dibangun dan diresmikan pada tanggal 31 Agustus 1921 oleh De Burgermeester Van Cheribon "J. H Johan", sehingga tanggal 31 Agustus 1921 ditetapkan sebagai hari lahir RSUD Gunung Jati Kota Cirebon. Pembangunan rumah sakit pada waktu itu dinilai sangat mewah dan mahal, biayanya adalah f.544.00,- (lima ratus empat puluh empat *gulden*) yang diperoleh dari Gemeente Van Cheribon ditambah dana dari pabrik gula sewilayah Cirebon serta dana para dermawan. Rumah sakit mulai berfungsi pada tanggal 1 September 1921 sebagai *Gemeemtelijk Ziekenhuis* dengan nama "*Oranje Ziekenhuis*", dibawah pimpinan dr. E. Gottlieb sebagai kepala rumah sakit yang pertama.

Rumah Sakit "*ORANJE*" pada saat itu mempunyai kapasitas 133 tempat tidur yang terdiri dari ruang direktur, ruang tata usaha, ruang apotek, ruang polikklinik, ruang laboratorium, ruang kamar bedah, ruang dapur, ruang cucian, ruang generator listrik, kamar mayat, ruang *zuster-huis*, ruang *hooftzuster-huis*, asrama putri, ruangan rawat dengan kapasitas 133 tempat tidur yang terbagi menjadi 7 tempat tidur kelas 1, 16 tempat tidur kelas 2, 24 tempat tidur kelas 3, 56 tempat tidur kelas 4, 16 tempat tidur untuk penyakit setengah menular dan 16 tempat tidur untuk penyakit menular. Data mengenai perkembangan selanjutnya antara tahun 1922-1929 didapat dari buku peringatan 50 Tahun Kota Besar Tjirbon, yang mengutarakan perkembangan jumlah hari perawatan dari 4 macam kelas perawatan dari tahun 1922 sampai 1929. Perkembangan selanjutnya antara tahun 1930 sampai dengan 1940 tidak banyak diketahui.

Menjelang pendudukan Jepang ada perubahan baik bentuk fisik maupun susunan ruangan yang disesuaikan dengan kebutuhan dan tingkat perkembangan pada waktu itu, antara lain diadakannya kamar bersalin, kamar rontgen/fisioterapi, asrama siswa kesehatan dan ruang administrasi. Pada tanggal 1 Maret 1942 seluruh rumah sakit beserta sarananya dievakuasi ke Rumah Sakit Sidawangi selama kurang lebih 2 minggu dan setelah kembali ke Kota Cirebon pada tanggal 15 Maret 1942, nama rumah sakit diubah dari Rumah Sakit *Oranje* menjadi Rumah Sakit Kesambi. Pada tanggal 8 Nopember 1975, nama rumah sakit diubah menjadi Rumah Sakit Gunung Jati Kelas D berdasarkan Surat Keputusan DPRD Kotamadya Daerah Tingkat II Cirebon Nomor : 30/DPRD/XI/75. Selanjutnya pada tanggal 22 Februari 1979 ditingkatkan menjadi Kelas C berdasarkan Surat Keputusan Menteri Kesehatan Republik Indonesia Nomor : 41/MENKES/SK/II/79 dan pada tanggal 21 Januari 1987 ditingkatkan lagi menjadi Rumah Sakit Kelas B berdasarkan Surat Keputusan Menteri Kesehatan Republik Indonesia Nomor : 41/MENKES /SK/I/87. Pada tanggal 30 Januari 1989 ditetapkan menjadi Rumah Sakit Umum Daerah Gunung Jati Kotamadya Daerah Tingkat II Cirebon Kelas B berdasarkan Surat Keputusan Menteri Dalam Negeri Nomor 061/350/SJ.

Dalam pengelolaan keuangan sejak tanggal 1 April 1996 dengan Peraturan Daerah Kotamadya Daerah Tingkat II Cirebon Nomor 15 Tahun 1995 ditetapkan sebagai “Unit Swadana Daerah”. Dalam upaya peningkatan pelayanan maka pada tahun 1997 berdasarkan Surat Keputusan Menteri Kesehatan Nomor : YM 02.03.3.5.5237. RSUD Gunung Jati Kota Cirebon ditetapkan dengan status “Akreditasi Penuh”. Pada tanggal 15 Februari 1998 berdasarkan rekomendasi dari Departemen Kesehatan melalui Surat Keputusan Menteri Kesehatan Nomor : 153/MENKES/SK/II/1998, RSUD Gunung Jati Kota Cirebon ditetapkan menjadi “Rumah Sakit Kelas B Pendidikan”. Peresmian sebagai Rumah Sakit Kelas B Pendidikan dilakukan oleh Gubernur Jawa Barat tanggal 21 April 1999 berdasarkan Surat Keputusan Mendagri Nomor : 445.03-1023 tanggal 12 November 1998 dengan Struktur Organisasi dan Tata Kerja RSUD Kelas B Pendidikan.

Seiring dengan perubahan paradigma penyelenggaraan otonomi daerah maka berdasarkan Peraturan Daerah Kota Cirebon Nomor : 5 Tahun 2002, RSUD Gunung Jati Kota Cirebon ditetapkan sebagai Lembaga Teknis Daerah. Berdasarkan Undang-Undang Nomor 1 Tahun 2004 tentang Badan Layanan Umum (BLU) dan Keputusan Walikota Nomor 445/Kep 359-DPPKD/2009, RSUD Gunung Jati Kota Cirebon resmi ditetapkan sebagai rumah sakit dengan Pola Pengelolaan Keuangan Badan Layanan Umum Daerah (PPK-BLUD) pada tanggal 14 Desember 2009. Pada tanggal 2 Agustus 2011, RSUD Gunung Jati Kota Cirebon dinyatakan LULUS dengan status akreditasi penuh 16 Kelompok Pelayanan oleh Komisi Akreditasi Rumah Sakit dengan mendapatkan Sertifikat KARS/SERF/40/VIII/2011 yang berlaku sampai dengan 2 Agustus 2014. Pada tanggal 10 Agustus 2011 Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon mendapatkan Penghargaan Rumah Sakit Berprestasi Program Rumah Sakit Sayang Ibu dan Bayi (RSSIB) Tingkat Provinsi Jawa Barat Tahun 2011.

Kemudian pada bulan Oktober 2012 Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon mendapatkan penghargaan sebagai Rumah Sakit Kelas B Pendiddikan Utama Klasifikasi Utama A.

2.1.2 Visi, Misi dan Tujuan

A. Visi

Visi dari Rumah Sakit Umum Daerah Gunung Jati adalah sebagai berikut: "Menjadikan Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon sebagai pusat pelayanan dan pendidikan kesehatan berkualitas se wilayah III pada tahun 2015".

B. Misi

Rumah Sakit Umum Daerah Gunung Jati memiliki misi sebagai berikut:

a. Meningkatkan mutu pelayanan Rumah Sakit.
b. Meningkatkan pengetahuan, kemampuan, dan keterampilan sumber daya Rumah Sakit baik medis, paramedis, maupun non medis.
c. Meningkatkan kualitas dan kuantitas sarana dan prasarana Rumah Sakit.
d. Meningkatkan fungsi Rumah Sakit sebagai sarana pendidikan.

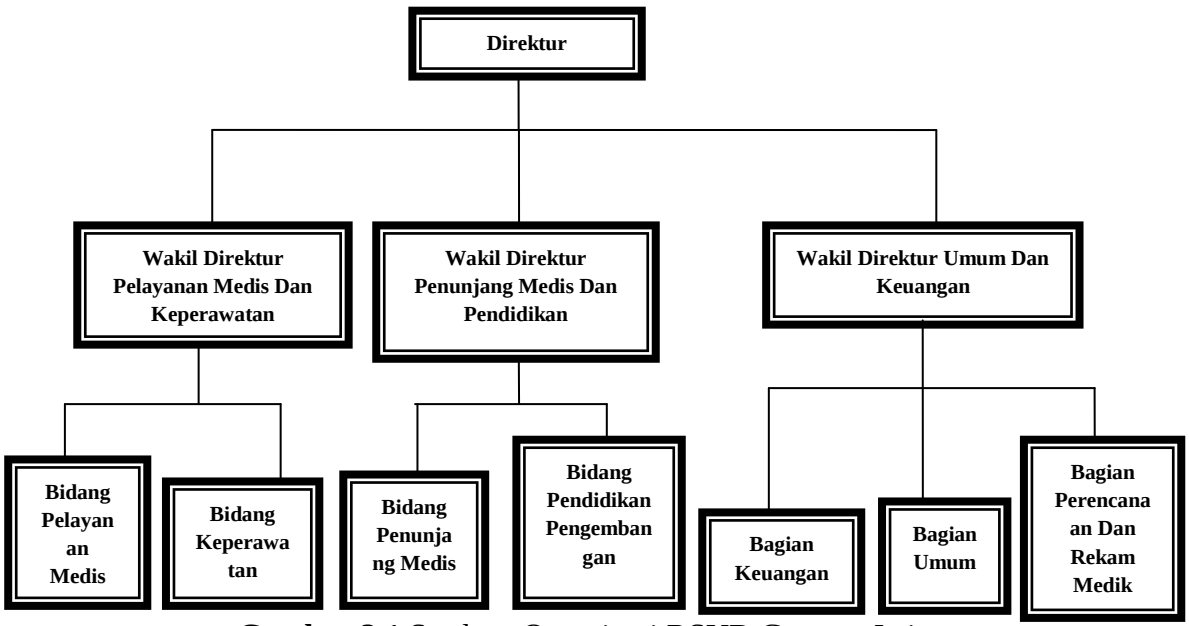
C. Tujuan

Tujuan dari Rumah Sakit Umum Daerah Gunung Jati adalah sebagai berikut:

a. Meningkatkan kelengkapan sarana dan prasarana sebagai penunjang peningkatan pelayanan yang berkualitas.
b. Memberdayakan kualitas, kuantitas, dan profesionalisme Sumber Daya Manusia.
c. Meningkatkan kesempurnaan terlaksananya Prosedur tetap Standar Operasional Pelayanan.
d. Meningkatkan status RSUD Gunung Jati Kota Cirebon sebagai Rumah Sakit Pendidikan Lanjutan.

2.1.3 Struktur Organisasi

Struktur organisasi dalam suatu institusi merupakan hal yang sangat penting, dengan adanya struktur organisasi ini memberikan pembagian tugas sesuai dengan bidangnya masing-masing. Adapun susunan organisasi yang ada pada RSUD Gunung Jati adalah sebagai berikut :



Gambar 2.1 Struktur Organisasi RSUD Gunung Jati

2.2 Landasan Teori

2.2.1 *Literature Review*

Telah dilakukan penelitian sebelumnya mengenai penerapan metode *FloydWarshall* dan membangun denah 3D perlu dilakukan studi pustaka sebagai salah satu dari penerapan. Beberapa *Literature review* tersebut adalah sebagai berikut:

1 Penelitian yang dilakukan oleh Rudy Adipranata, Liliana, dan Yohanes Pribadi berjudul, "PEMBUATAN PERANGKAT LUNAK EDITOR DENAH RUANG 2D UNTUK DIVISUALISASIKAN SECARA 3D". Penelitian ini menjelaskan tentang merancang rancangan denah dalam bentuk 2D dan memvisualisasikannya secara 3D.

2 Penelitian yang dilakukan oleh Raden Aprian Diaz Novandi berjudul, "Perbandingan Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall* dalam Penentuan Lintasan Terpendek (*Single Pair Shortest Path*)". Penelitian ini menjelaskan tentang perbandingan algoritma djikstra dan algoritma *Floyd Warshall* dalam menentukan keakurasian untuk memperoleh hasil pencarian rute terpendek yang tepat dan akurat.

3 Penelitian yang dilakukan oleh Rio Andreanto, TeguhSutanto, M.Kom.,MCP dan Vicky M Taufik, SE.Ak.,S.Kom berjudul, APLIKASI PEMANDU WISATA KEBUN BINATANG SURABAYABERBASIS ANDROID. Penelitian ini menjelaskan tentang membangun aplikasi pemandu berbasis android untuk mempermudah pengunjung dalam memperoleh informasi kebun binatang.

2.2.2 Android

2.2.2.1 Pengertian Android

Android adalah sekumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android *Standart Development Kid* (SDK) menyediakan perlengkapan dan *Application Programming Interface* (API) yang diperlukan untuk

mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*.

Android dikembangkan oleh Google bersama *Open Handset Alliance* (OHA) yaitu aliansi perangkat seluler terbuka yang terdiri dari 47 perusahaan *Hardware*, *Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

2.2.2.2 Sejarah Perkembangan Android

Android, Inc. didirikan di *Palo Alto, California*, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger),[29] Rich Miner (pendiri *Wildfire Communications, Inc*.), Nick Sears (mantan *VP T-Mobile*), dan Chris White (kepala desain dan pengembangan antarmuka WebTV) untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya". Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi *Symbian* dan *Windows Mobile* (*iPhone Apple* belum dirilis pada saat itu). Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman, *Android Inc*. dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler.

Google mengakuisisi *Android Inc.* pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri *Android Inc*. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Setelah itu, tidak banyak yang diketahui tentang perkembangan *Android Inc*., namun banyak anggapan yang menyatakan bahwa Google berencana untuk memasuki pasar telepon seluler dengan tindakannya ini. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan *kernel Linux*. Google memasarkan *platform* tersebut kepada produsen perangkat seluler dan operator nirkabel,

dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui.Google telah memilih beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerjasama ini terbuka bagi siapapun yang ingin berpartisipasi.

Android pertama kali diluncurkan pada 5 November 2007, dan *smartphone* pertama yang menggunakan sistem operasi Android dikeluarkan oleh *T-Mobile* dengan sebutan G1 pada bulan September 2008.Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.1 dirlis pada 9 maret 2009, versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, versi 2.0 dirilis pada 26 Oktober 2009 diberi nama *Éclair*, versi 2.2 dirilis pada 20 Mei 2010 diberi nama *Froyo (Frozen Yoghurt)*, versi 2.3 dirilis pada 6 Desember 2010 diberi nama *Gingerbread*, versi 3.0 dirilis pada Mei 2011 diberi nama *Honeycomb*, versi 4.0 dirilis pada 19 Oktober 2011 diberi nama ICS (*Ice Cream Sandwich*).

2.2.3 Java

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program *java* tersusun dari bagian yang disebut kelas.Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram *Java* banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*).*Java* API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih.Jadi ada dua hal yang harus dipelajari dalam *Java*, yaitu mempelajari bahasa *Java* dan bagaimana mempergunakan kelas pada *Java API*. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada *Java* program *javac* untuk mengkompilasi file kode sumber *Java* menjadi kelas-kelas *bytecode*.

File kode sumber mempunyai ekstensi **.java*.*Kompilator javac* menghasilkan *file bytecode* kelas dengan ekstensi *.class.*Interpreter* merupakan modul utama sistem *Java* yang digunakan aplikasi *Java* dan menjalankan program *bytecode Java*.

2.2.3.1 Java Versi Lama (Java 1)

Pada awal perilisannya, versi *Java* masih disebut dengan JDK (*Java Development kita*). Dalam JDK, semua kebutuhan untuk pengembangan program dan eksekusi program masih tergabung jadi satu.[3] Penamaan ini berlaku sampai *Java 1.1*. namun sekarang, setelah *Java 1.2*, *Sun Microsystem* menamainya dengan JSDK (*Java Software Development Kit*) dalam hal ini kebutuhan untuk pengembangan program dipisahkan dengan kebutuhan kesekusi. Bagian *software* yang digunakan untuk kebutuhan eksekusi program disebut dengan JRE (*Java-Runtime Environment*).Selanjutnya, *Java* 1.2 disederhanakan penamaanya menjadi "*Java 2*".

2.2.3.2 Java Versi Baru (Java 2)

Sun Microsystems telah mendefinisikan tiga buah edisi dari *Java 2*, yaitu sebagai berikut :

a. *Java 2 Standard Edition(J2SE),* yang digunakan untuk mengembangkan aplikasi-aplikasi desktop dan applet (aplikasi *Java* yang dapat dijalankan di dalam *browser web*).
b. *Java 2 Enterprise Edition* (J2EE), merupakan *superset* dari J2SE yang memperbolehkan untuk mengembangkan palikasi-aplikasi berskala besar (*enterprise*), yaitu dengan melakukan pembuatan aplikasi-aplikasi di sisi server dengan menggunakan EJBs (*Enterprise JavaBeans*), aplikasi web dengan menggunakan *Servlet* dan JSP (*JavaServer Pages*) dan teknologi lainnya serperti CORBA (*Common Object Request Broker Architecture*) dan XML (*Extensible Markup Language*).
c. *Java 2 Micro Edition* (J2ME), merupakan subset dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh.

2.2.4 Database

Database dapat diartikan sebagai kumpulan data yang terdiri atas satu atau lebih tabel yang terintegrasi satu sama lain, di mana setiap pemakai (*user*) diberi wewenang (otorisasi) untuk dapat mengakses (mengubah, menghapus, menganalisis, menambah, memperbaiki) data dalam tabel-tabel tersebut.

2.2.4.1 Fungsi Database

Adapun fungsi database diantaranya adalah sebagai berikut :

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menentukan informasi.
2. Menentukan kualitas informasi. Informasi dapat dikatakan bernilaiapabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
3. Mengurangi duplikasi data (data *redundancy*).
4. Hubungan data dapat ditingkatkan (data *relatability*).
5. Mengurangi pemborosan tempat simpanan luar.

2.2.4.2 Kriteria Database

Dari penjelasan sebelumnya dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data *oriented* dan bukan program *oriented*
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah dapat digunakan dengan cara-cara yang berbeda.

2.2.5 3 Dimensi

3D atau 3 Dimensi adalah sebuah objek atau ruang yang memiliki panjang, lebar dan tinggi yang memiliki bentuk.3D tidak hanya digunakan dalam matematika dan fisika saja melainkan dibidang grafis, seni, animasi, komputer

dan lain-lain. Konsep tiga dimensi atau 3D menunjukkan sebuah objek atau ruang memiliki tiga dimensi geometris yang terdiri dari: kedalaman, lebar dan tinggi. Contoh tiga dimensi suatu objek atau benda adalah bola, piramida atau benda spasial seperti kotak sepatu. Istilah "3D" juga (dan salah) yang digunakan (terutama bahasa Inggris) untuk menunjukkan representasi dalam grafis komputer (*digital*), dengan cara menghilangkan gambar *stereoscopic* atau gambar lain dalam pemberian bantuan, dan bahkan efek *stereo* sederhana, yang secara konstruksi membuat efek 2D (dalam perhitungan proyeksi perspektif, *shading*)

2.2.5.1 Karakteristik 3D

Mengacu pada tiga dimensi spasial, bahwa 3D menunjukkan suatu titik koordinat *Cartesian* X, Y dan Z. Penggunaan istilah 3D ini dapat digunakan di berbagai bidang dan sering dikaitkan dengan hal-hal lain seperti spesifikasi kualitatif tambahan (misalnya: grafis tiga dimensi, 3D video, film 3D, kacamata 3D, suara 3D). Istilah ini biasanya digunakan untuk menunjukkan relevansi jangka waktu tiga dimensi suatu objek, dengan gerakan perspektif untuk menjelaskan sebuah "kedalaman" dari gambar, suara, atau pengalaman taktil. Ketidak jelasan istilah ini menentukan penggunaannya dalam beberapa kasus yang tidak jelas juga yaitu penggunaannya tidak hanya pada contoh-contoh diatas melainkan (sering dalam iklan dan media).

2.2.5.2 3D dalam Komputer

Saat ini 3D digambarkan untuk mensimulasikan perhitungan berdasarkan layar proyeksi dua dimensi dan efek tiga-dimensi seperti monitor komputer atau televisi. Perhitungan ini memerlukan beban pengolahan besar sehingga beberapa komputer dan *konsol* memiliki beberapa tingkat percepatan grafis 3D untuk perangkat yang dikembangkan untuk tujuan ini. Komputer memiliki kartu grafis panggilan atau tambahan untuk meningkatkan akselerasi 3D. Perangkat ini dibentuk dengan satu atau lebih *prosesor* (GPU) yang dirancang khusus untuk mempercepat perhitungan yang melibatkan tiga dimensi gambar yang

mereproduksi pada layar dua dimensi dan dengan melepaskan beban pengolahan pada CPU atau *central processing unit computer*.

Dalam komputasi, model tiga dimensi (angka atau grafis) dibuat tanpamembutuhkan perhitungan yang sangat kompleks, tetapi sangat banyak.3D dapat direpresentasikan baik oleh prospek dari berbagai arah pada layar dua dimensi (yang membuat istilah "3D" tidak benar, layar dengan hanya dua dimensi), atau pada jenis perangkat atau kacamata film yang timbul dari LCD untuk melihat gambar yang berbeda pada setiap pandangan mata.

Sejak akhir 1990-an, banyak komputer yang memiliki *prosesor* yang didedikasikan untuk melampiran jenis perhitungan (*graphics processing* unit atau GPU). Beberapa paket perangkat lunak, termasuk *Blender* untuk membuat model 3D dengan komputer dan hasilnya disebut dengan gambar 3D sintesis.*Software* untuk membuat 3D biasanya yaitu *Autodesk Maya* atau *Blender* 3D. dan *software* untuk membuat bangunan 3D *modelling* yaitu *Autocad*, *3DMax*, *SketchUp* dll.

2.2.6 Definisi Graf

Graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang terhubung oleh sisi (*edge*) atau busur (*arc*).Biasanya *graf* digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur). Suatu *graf* G didefenisikan sebagai pasangan himpunan (V,E), di mana V = himpunan yang berisikan simpul pada *graf* tersebut $\{V_1,V_2,..,V_3\}$ dan E adalah himpunan sisi yang menghubungkan simpul-simpul $\{e_1, e_2,.., e_3\}$ atau dapat ditulis dengan notasi G = (V,E).

Ada terdapat beberapa jenis *graf* yang bisa kita gunakan, yaitu beberapa diantaranya adalah sebagai berikut :

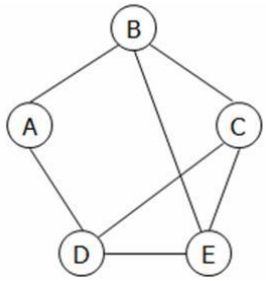
A. *Graf* Berarah : adalah *graf* yang *edge*-nya memiliki arah, sebagai contoh *edge* ABmenghubungkan *vertex* A ke B, dimana hubungan *vertex* B ke A, harus diperoleh dari*edge* lain, yaitu *edge* BA, dan jika *edge* BA tidak ada, maka *vertex* B ke A tidak memiliki hubungan, meski *vertex* A ke B memiliki hubungan.

B. *Graf* Tak Berarah : adalah graf yang edge-nya tidak memiliki arah, sehingga jika *edge*AB menghubungkan *vertex* A ke B, maka secara otomatis juga menghubungkan *vertex* B ke A.
C. *Graf*Berbobot : adalah suatu *graf* dimana *edge* dari *graf* tersebut memiliki bobot ataunilai tertentu.
D. *Graf* Tidak Berbobot : adalah suatu *graf* dimana *edge* dari *graf* tersebut tidak memilikibobot atau nilai. Untuk merepresentasikannya dalam pemrograman komputer, *graf* dapat disusun dari *LinkedList* yang berada dalam *LinkedList*.

2.2.6.1 Representasi Graph dalam Bentuk Matriks

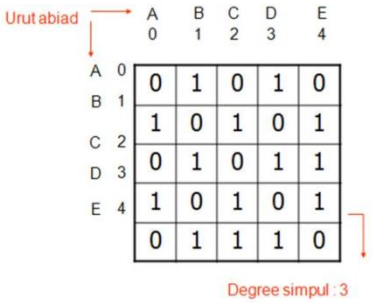
Matriks dapat digunakan untuk menyatakan suatu *graf*, *graf* yang direpresentasikan pada matriks, dapat dibedakan sebagai berikut :

a. *Graph* Tak Berarah



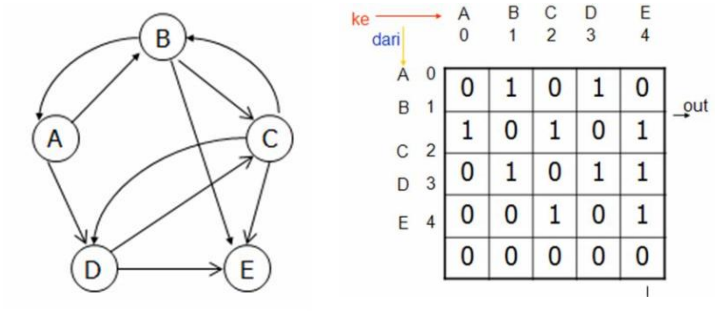
Gambar 2.2 *Graph* tak berarah

Graf tersebut dapat direpresentasikan dalam sebuah matriks5x5 , dimana baris dankolom di matriks tersebut menunjukan *vertex* yang ada.



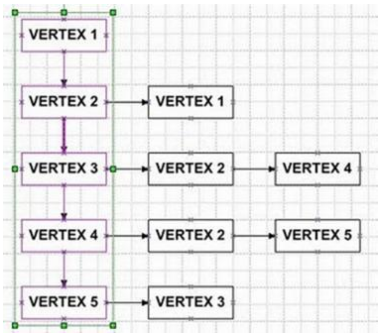
Gambar 2.3 *Graph* Tak Berarah Dalam Bentuk Matriks 5x5

b. *Graph* Berarah



Gambar 2.4 *Graph* Berarah dan Bentuk Matrik

Dalam matrik diatas dapat kita lihat bahwa kotak yang berisi angka satu menunjukan bahwa dalam dua *vertex* tersebut terdapat*edge* yang menghubungkannya.Dan jika dalam kotak terdapat angka nol, maka hal tersebut menandakan tidak ada *edge* yang mengubungkan secara langsung dua *vertex* tersebut.Untuk representasi dalam pemorgraman komputer, graf tersebut dapat digambarkan seperti dibawah ini :



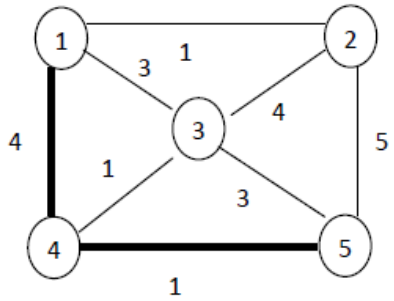
Gambar 2.5 *Graph* Representasi Pemrograman Komputer

2.2.6.2 Definisi Lintasan (Path)

Lintasan adalah hubungan antar titik atau *node* dalam sebuah *graf*. Suatu lintasan yang berawal dan berakhir pada *node* yang sama, maka disebut lintasan tertutup (*close path*), jika node awal dan *node* akhir dari lintasan tersebut berbeda, disebut lintasan terbuka (*open path*).

2.2.6.3 Lintasan Terpendek (Shortest Path)

Lintasan terpendek adalah lintasan yang memiliki total bobot minimum untuk mencapai suatu tempat dari tempat tertentu. Lintasan terpendek dapat dicari dengan menggunakan *graf.Graf* yang digunakan adalah graf yang berbobot, yaitu *graf* yang setiap sisinya diberikan suatu nilai atau bobot.Bobot pada sisi *graf* dapat menyatakan, waktu, biaya dan sebagainya.



Gambar 2.6 Lintasan Terpendek Pada *Graph*

Gambar 2.6 menunjukkan lintasan terpendek (garis tebal) dari *node* 1 ke *node* 5 dengan jalur 1-4-5 dengan jumlah bobot lima (5).

2.2.7 Algoritma Floyd Warshall

Algoritma *Floyd Warshall* ada1ah salah satu algoritma dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi- solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelurnnya dan ada kemungkinan lebih dari satu solusi (Thomas H. Cormen, 2003, p.629).

Algoritma *Floyd Warshall* juga membandingkan semua kemungkinan lintasan pada *graf* untuk setiap sisi dari semua simpu1. Algoritma *Floyd Warshall* menerapkan pemrograrnan dinarnis sehingga lebih menjarnin keberhasilan penemuan solusi optimum untuk kasus penemuan lintasan terpendek (*single pair shortest path*).

Dalam proses penye1esaian menggunakan program dinamis, pendekatan yang dilakukan bisa jadi ada dua macam, yaitu pendekatan maju

(*forward*) dan mundur (*backward*), dan perlu untuk diketahui pula bahwa solusi yang dihasilkan dari kedua pendekatan itu adalah sama. Solusi dari program dinamis bisa jadi lebih dari satu macam (Thomas H. Cormen, 2003, p.323).

2.2.7.1 Analisis Algoritma Floyd-Warshall

Algoritma *Floyd Warshall* membandingkan semua kemungkinan lintasan pada grafuntuk setiap sisi dari semua simpul. Menariknya, algoritma ini mampu mengerjakan proses perbandingan ini sebanyak V 3 kali (bandingkan dengan kemungkinan jumlah sisi sebanyak V 2 (kuadrat jumlah simpul) pada *graf,* dan setiap kombinasi sisi diujikan). Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal.

Misalkan terdapat suatu graf G dengan simpul-simpul V yang masing-masing bernomor 1 s.d. N (sebanyak Nbuah). Misalkan pula terdapat suatu fungsi shortestPath(i, j, k) yang mengembalikankemungkinan jalur terpendek dari i ke j dengan hanyamemanfaatkan simpul 1 s.d. k sebagai titik perantara.

Perlu diketahui bahwa jalur terpendek dari i ke j yanghanya melewati simpul 1 s.d. k telah didefinisikan padafungsi shortestPath(i, j, k) dan telah jelas bahwa jika adasolusi dari i s.d. k+1 hingga j, maka panjang dari solusitadi adalah jumlah (konkatenasi) dari jalur terpendek dari i s.d. k+1 (yang melewati simpul-simpul 1 s.d. k), dan jalurterpendek dari k+1 s.d. j (juga menggunakan simpul-simpuldari 1 s.d. k).

Maka dari itu, rumus untuk fungsi shortestPath(i,j, k) bisa ditulis sebagai suatu notasi rekursif sbb.:

Tabel 2.1 Rumus Fungsi *shortestPath*

```
Basis-0
shortestPath(i, j, 0) = edgeCost(i,
j);
Rekurens
shortestPath(i, j, k) =
min(shortestPath(i, j, k-1),
```

```
shortestPath(i, k, k-1) +
shortestPath(k, j, k-1));
```

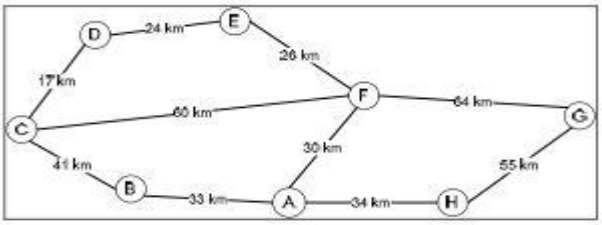
Rumus ini adalah inti dari algoritma *Floyd-Warshall*.Algoritma ini bekerja dengan menghitung*shortestPath*(i,j,1) untuk semua pasangan (i,j), kemudianhasil tersebut akan digunakan untuk menghitung*shortestPath*(i,j,2) untuk semua pasangan (i,j), dst. Prosesini akan terus berlangsung hingga k = n dan kita telahmenemukan jalur terpendek untuk semua pasangan (i,j) menggunakan simpul-simpul perantara.

Tabel 2.2 *Psoudcode* Algoritma *Floyd Warshal*

```
//Asumsikan bahwa terdapat fungsi
edgeCost(i,j) yang mengembalikan biaya
(cost) di ujung dari i ke j (tak hingga
jika tidak ada)
//Juga asumsikan bahwa n adalah jumlah
simpul dan edgeCost(i,i)=0
path = array of integer;
//Matriks dua dimensi. Pada setiap langkah
di algoritma, path[i][j] adalah jalur
terpendek dari i ke j memanfaatkan nilai
perantara pada (1..k-1). Setiap path[i][j]
diinisialisasi ke edgeCost(i,j);
procedure FloydWarshall()
k traversal [1..n]
foreach (i,j) pada [1..n]
path[i][j] = min(path[i][j],
path[i][k]+path[k][j]);
{end foreach}
{end traversal}
```

2.2.6.2 Contoh Penerapan Algoritma Floyd Warshall:

Misalkan terdapat suatu *graf* berbobot yangmerepresentasikan kondisi keterhubungan antarkota disuatu daerah, dengan ilustrasi sebagai berikut.



Gambar 2.7. Representasi keterhubungan antarkota dalam*graf* berbobot

Misalkan seseorang akan melakukan perjalanan darikota A ke kota C. Orang tersebut mencoba untukmenerapkan algoritma *Floyd-Warshall* untuk mencari jalur terpendek dari kota A kekota C.

Sekarang, orang tersebut mencoba menerapkan algoritma*Floyd-Warshall* dengan pendekatan pemrograman dinamismaju (*forward*).

Basis

$f1$ (s) = cx1s

Rekurens

fk (s) = min xk {cxks + fk-1(xk)}, k = 2, 3, 4

Tahap 1:

$f1$ (s) = cx1s

Tabel 2.3 Iterasi Awal

s	Solusi optimum	
	f_1 (s)	x_1
B	33	A
F	30	A
H	34	A

Tahap 2:

$f2$ (s) = min $s2$ {cx2s + f1(x2)}

Tabel 2.4 Iterasi kedua

x_2 / s	$f_2(x_2, s) = cx_2s + f_1(x_2)$			Solusi optimum	
	B	F	H	$f_2(s)$	x_2
C	74	90	∞	74	B
E	∞	56	∞	26	F
G	∞	94	89	89	H

Dari hasil pencarian jalur terpendek dari A ke Cmenggunakan algoritma *Floyd-Warshall* (pemrogramandinamis), ditemukan bahwa jarak terpendek dari A ke Cadalah **74** km dengan jalur (A – B – C).

2.2.8 Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata.Suatu perusahaan *software* yaitu *Rational Software*, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD).

2.2.8.1 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation

dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, *Java*, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memilikimakna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.2.8.2 Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*)
4. antar objek.
5. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
6. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.
7. *Activity Diagram* untuk memodelkan perilaku userdan objek di dalam sistem.
8. *Class Diagram* untuk memodelkan struktur kelas.
9. *Objek Diagram* untuk memodelkan struktur objek.
10. *Component Diagram* untuk memodelkan komponen objek.
11. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, sequence diagram, collaboration diagram,* dan*class diagram.*

2.2.8.3 Tujuan UML

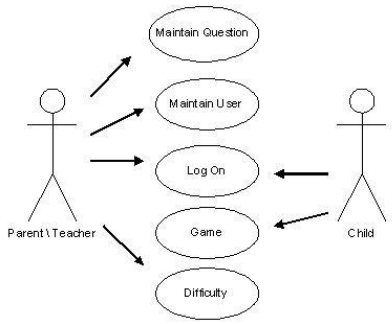
Tujuan utama UML diantaranya untuk :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam bahasa pemodelan.

2.2.8.4 Use Case Diagram

"Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem."[4]*Use case diagram* terdiri atas *diagram* untuk *use case* dan *actor*. *Actor*merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*.*Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case.*

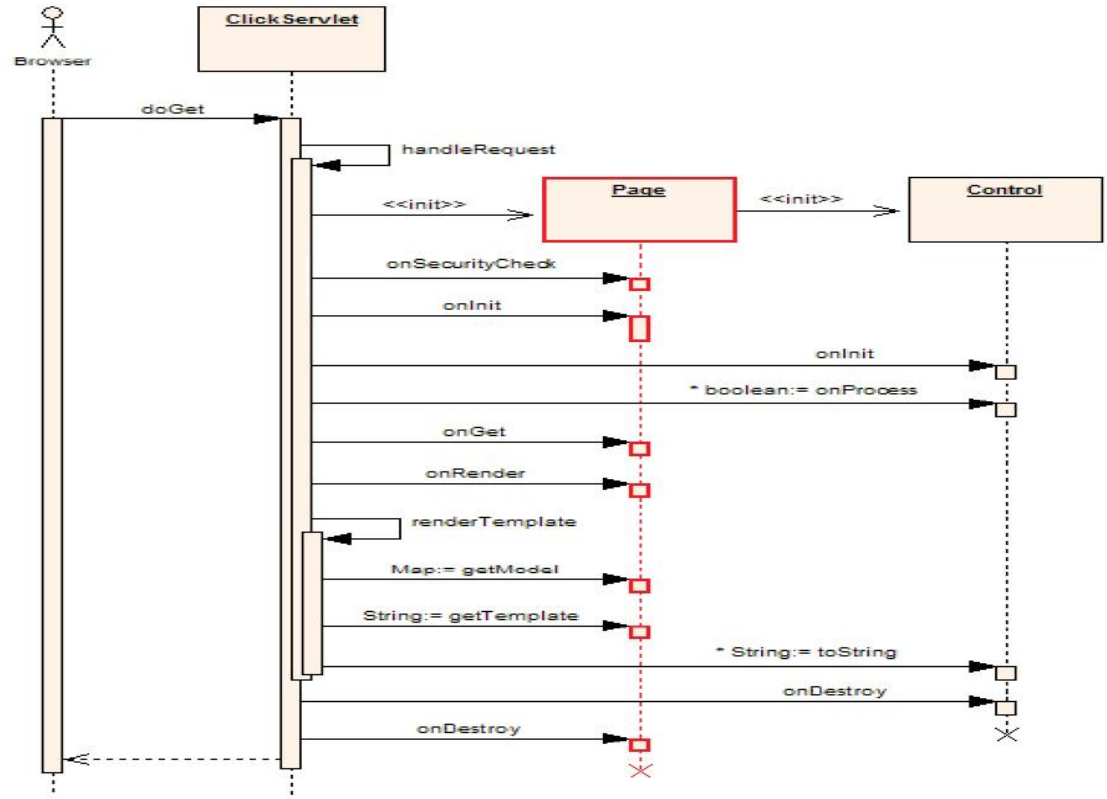


Gambar 2.8 *Use CaseDiagram*

2.2.8.5 Sequence Diagram

Sequence diagram (diagramurutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan atau *message*.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau even untuk menghasilkan *output* tertentu. *Sequence Diagram* diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

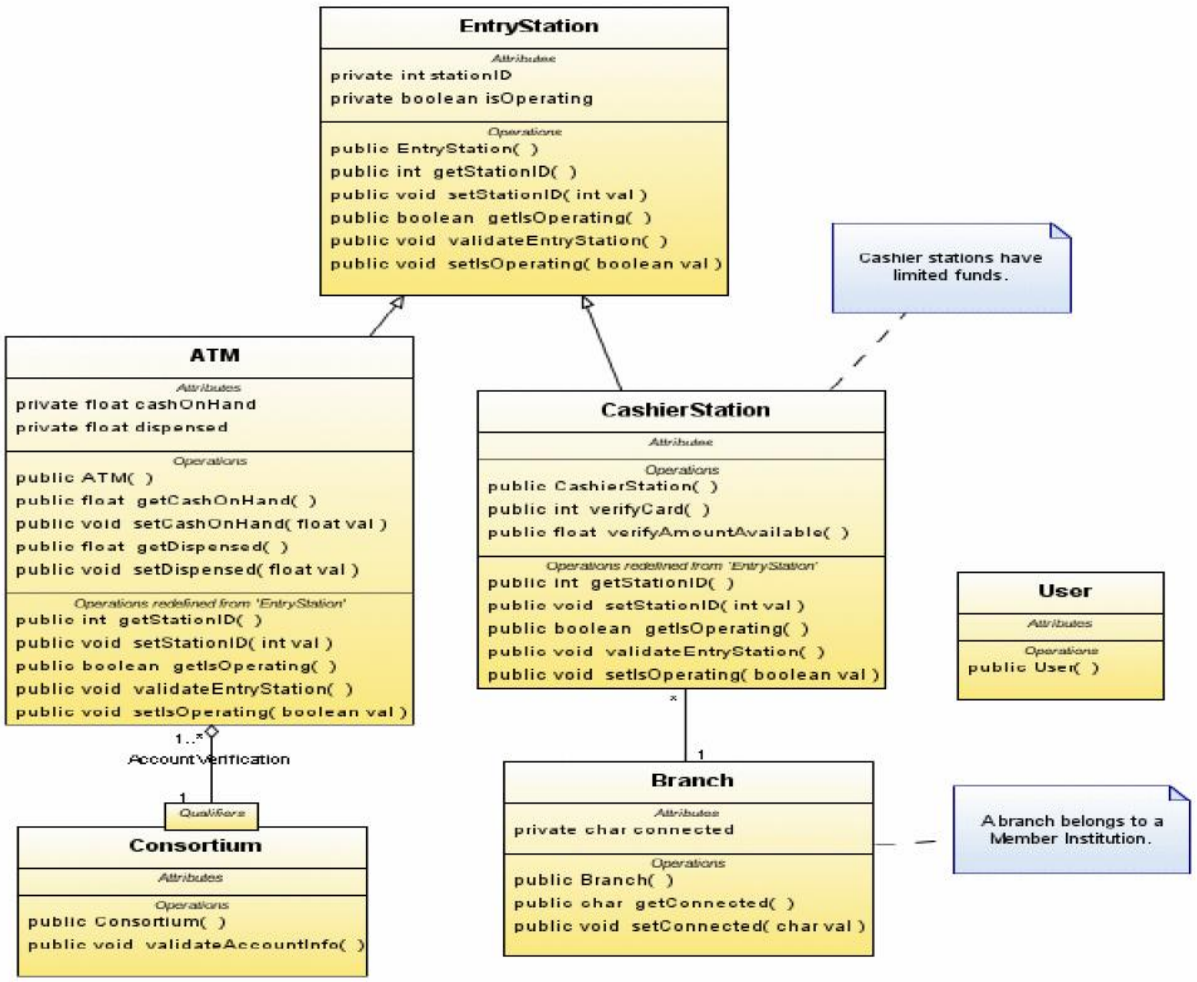


Gambar 2.9 *Sequence Diagram*

2.2.8.6 Class Diagram

Class Diagram adalah diagram yang menunjukan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram

merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML (Henderi, 2008**).** "Sementara menurut (Whitten L. Jeffery et al 2004:432) class diagram adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukan *class-class* objek yang menyusun sebuah sistem dan juga hubungan antara *class* objek tersebut."[4]



Gambar 2.10 *Class Diagram*