BAB II
LANDASAN TEORI

2.1 Profil PT. Jasa Marga (persero)

2.1.1 Sejarah Instansi

Jasa Marga merupakan perusahaan perintis penyelenggaraan jalan tol di Indonesia, yang didirikan pada tanggal 01 Maret 1978. Sebagai jalan tol pertama di Indonesia yang dioperasikan oleh Jasa Marga, Jalan tol Jagorawi (Jakarta-Bogor-Ciawi) merupakan tonggak sejarah bagi perkembangan industri jalan tol di Tanah Air. Berbekal pengalaman selama lebih dari tiga dasawarsa, Perseroan membuktikan kepiawaiannya dengan tetap menjadi pemimpin pasar di industri jalan tol di Tanah Air. Hingga saat ini Perseroan telah mengoperasikan 531 km jalan tol atau 72 % dari total panjang jalan tol di Indonesia[1].

Perseroan berhasil memenangkan 3 (tiga) konsesi baru yaitu Bogor Ring Road, Semarang-Solo, dan Gempol-Pasuruan pada tahun 2004 serta 2 (dua) ruas JORR 2 yaitu Cengkareng-Kunciran dan Kunciran-Serpong pada tahun 2007. Tiga ruas tol baru lainnya yaitu Surabaya-Mojokerto, JORR W2, serta Gempol-Pasuruan juga menambah jumlah ruas tol yang saat ini dimiliki Perseroan. Sebanyak 8 (delapan) ruas tol baru dengan panjang sekitar 200 km yang saat ini sedang dipersiapkan Perseroan tersebut diharapkan dapat beroperasi secara bertahap antara 2011-2013.

Perseroan telah melalui berbagai peristiwa penting dan perubahan dalam perjalanannya. Pada awal berdirinya, Perseroan berperan tidak hanya sebagai operator, tetapi juga memikul tanggung jawab sebagai otoritas jalan tol di Indonesia. Tahun 2004, peran otorisator dikembalikan kepada Pemerintah dengan dikeluarkannya Undang Undang No. 38 tahun 2004 tentang Jalan. Peran otorisator dilaksanakan oleh Badan Pengatur Jalan Tol (BPJT). Sebagai konsekuensinya, Jasa Marga menjalankan fungsi sepenuhnya sebagai sebuah perusahaan pengembang dan operator jalan tol dengan berorientasi pada kaidah-kaidah korporasi.

Perubahan ini mendorong Perseroan untuk lebih fokus dalam mengembangkan bisnis jalan tol, mulai dari perencanaan, pembangunan hingga

pengoperasian jalan tol. Perseroan pun semakin mendapatkan kepercayaan pemangku kepentingan terutama investor karena Perseroan dapat lebih berkonsentrasi dalam meningkatkan nilai Perseroan.

2.1.2 Logo Instansi

Logo PT.Jasa Marga(Persero) dapat dilihat pada gambar II.1



Gambar II.1 Logo Instansi

Inti dari logo tersebut adalah semangat dan profesionalisme yang lebih modern, simpel, efisien dan berorientasi pada teknologi baru, serta dapat menjawab tantangan persaingan industri global, tanpa meninggalkan warisan pengalaman dan pengetahuan yang telah dimilikinya.

2.1.3 Visi Misi Instansi

Visi

Visi Menjadi Perusahaan modern dalam bidang pengembangan dan pengoperasian jalan tol, menjadi pemimpin (*leader*) dalam industri jalan tol dengan mengoperasikan mayoritas jalan tol di Indonesia, serta memiliki daya saing yang tinggi di tingkat Nasional dan Regional.

Misi

Menambah panjang jalan tol secara berkelanjutan, sehingga Perusahaan menguasai paling sedikit 50% panjang jalan tol di Indonesia dan usaha terkait lainnya, dengan memaksimalkan pemanfaatan potensi keuangan Perusahaan serta meningkatkan mutu dan efisiensi jasa pelayanan jalan tol melalui penggunaan teknologi yang optimal dan penerapan kaidah-kaidah manajemen Perusahaan modern dengan tata kelola yang baik.

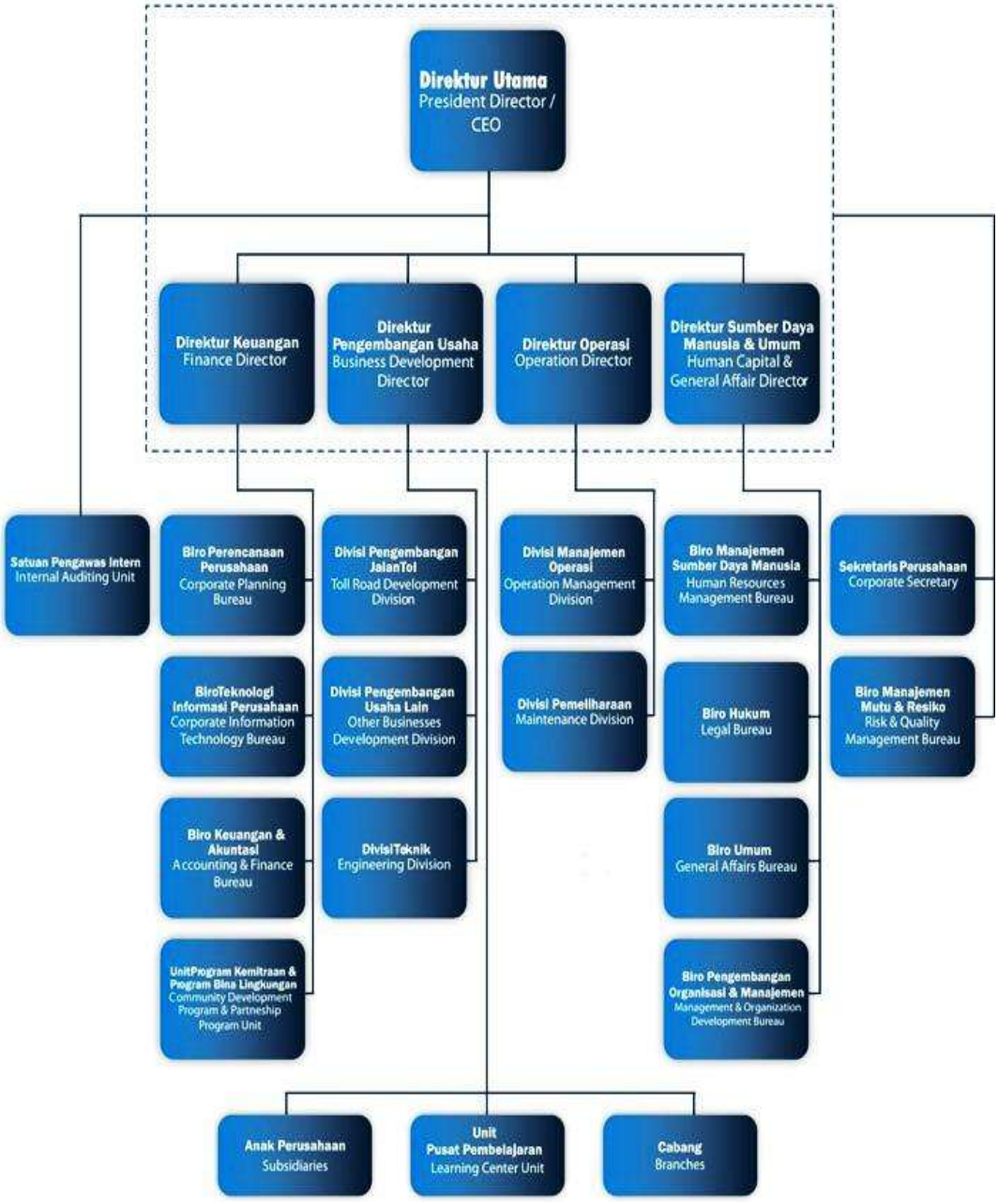
2.1.4 Badan Hukum Instansi

Jasa Marga berdiri berdasarkan akta nomor 1 tanggal 1 maret 1978 yang di buat dihadapan notaris Kartini Muljadi,SH., dengan nama "PT.Jasa Marga (*Indonesia Highway Corporation*)", kemudian berdasarkan akta nomor 187 tanggal 19 mei 1981 masih dengan notaris yang sama, nama perseroan berubah menjadi "PT. Jasa Marga (Persero)" dan telah memperoleh pengesahan dari menteri kehakiman republik Indonesia dengan keputusan nomor Y.A.5/130/1 tertanggal 22 Februari 1982 dan di daftarkan dalam buku register di kantor pengadilan negeri Jakarta berturut-turut dibawah nomor 766 dan 767 tanggal 2 maret 1982 serta di umumkan dalam Berita Negara Republik Indonesia No 73 tanggal 10 september 1982, tambahn nomor 1138 (untuk selanjutnya akta no 1 tanggal 1 maret 1978 dan akta no 187 tanggal 19 mei 1981 tersebut disebut "Akta Pendirian"). Pendirian perseroan tersebut sesuai dengan ketentuan-ketentuan yang terdapat dalam UU no 9 tahun 1969 tentang Penetapan Peraturan Pemerintahan Pengganti UU no 1 tahun 1969 tentang Bentuk-bentuk Usaha Negara menjadi UU, PP no 12 tahun 1969 tentang perusahaan Jasa Marga (Persero) dan PP no 4 tahun 1978 tentang pernyataan modal Negara Republik Indonesia dalam pendirian perusahaan Jasa Marga (Persero) di bidang pengelolaan, pemeliharaann dan pengadaan jaringan Jalan Tol serta surat Keputusan Menteri Keuangan Repulik Indonesia no 90/KMK.06/1978 tanggal 27 Februari 1978 tentang penetapan modal perusahaan Jasa Marga (Persero) di bidang jalan tol.

Dalam rangka penawaran umum perdana saham kepada masyarakat, anggaran dasar perseroan diubah berdasarkan akta pernyataan keputusan rapat no 27 tanggal 12 september 2007 yangdibuat di hadapan notaris Ny. Poerbaningsih Adi Warsito SH. Dalam akta tersebut nama peseroan di ubah menjadi "Perusahaan Perseroan (Persero) PT.Jasa Marga (Indonesia Highway Corporatama)Tbk." atau disingkat "PT.Jasa Marga (Persero)Tbk". Perubahan anggaran dasar perseroan tersebut telah memperoleh persetujuan dari Menteri Hukum dan Hak Asasi Manusia Republik Indonesia berdasarkan keputusan no W7-10487HT.01.04-TH 2007 tanggal 21 september 2007 dan pemberitaan

atas perubahan anggaran dasar tersebut telah diterima dan dicatat dalam database disisminbakum Departeman Hukum dan Hak Asasi Manusia sebagai mana tertera dalam surat Kepala Kanwil DKI Jakarta a.n.Menteri Hukum dan Hak Asasi Manusia nomor W7-HT.0110-13313 tanggal 24 september 2007 dan telah di daftarkan dalam daftar perusahaan di kantor pendaftaran perusahaan kodya Jakarta Timur nomor 269/ RUB/09.04/X/07 tanggal 4 oktober 2007.

2.1.5 Struktur Organisasi



Gambar II.2 Struktur Organisasi

2.1.6 Bidang Usaha PT. Jasa Marga

Bidang Usaha Jalan Tol

Bidang usaha Jasa Marga adalah membangun dan menyediakan jasa pelayanan jalan tol. Untuk itu Jasa Marga melakukan aktifitas usaha sebagai berikut:

1. Melakukan investasi dengan membangun jalan tol baru.
2. Mengoperasikan dan memelihara jalan tol.
3. Mengembangkan usaha lain, seperti tempat istirahat, iklan, jaringan serat optik dan lain-lain, untuk meningkatkan pelayanan kepada pemakai jalan dan meningkatkan hasil usaha perusahaan.
4. Mengembangkan usaha lain dalam koridor jalan tol.

Saat ini Jasa Marga mengelola dan mengoperasikan 13 hak pengusahaan (konsesi) jalan tol melalui sembilan kantor cabang dan satu anak perusahaan yaitu:

1. Jalan tol Jagorawi
2. Jalan Tol Jakarta-Tangerang
3. Jalan Tol Jakarta- Cikampek
4. Jalan Tol Dalam Kota Jakarta
5. Jalan Tol Prof. Dr.Ir. Sedyatmo
6. Jalan Tol Serpong-Pondok Aren (dioperasikan oleh JLJ)
7. Jalan Tol Cikampek -Purwakarta-Cileunyi
8. Jalan Tol Padalarang –Cileunyi
9. Jalan Tol Palimanan-Kanci
10. Jalan Tol Semarang
11. Jalan Tol Surabaya Gempol
12. Jalan Tol Belawan-Medan-Tanjung Morawa
13. Jalan Tol Lingkar Luar Jakarta (dioperasikan oleh JLJ)

Dibawah ini adalah anak perusahaan Jasa Marga pemegang konsesi Jalan Tol :

a. PT Jalan Tol Lingkar Luar Jakarta (JLJ) Jalan Tol JORR kepemilikan saham sebesar 99%.

b. PT Marga Sarana Jabar (MSJ) Jalan Tol Bogor Ring Road kepemilikan saham sebesar 55% .
c. PT Trans Marga Jateng (TMJ) Jalan Tol Semarang-Solo kepemilikan saham sebesar 60% .
d. PT Marga Trans Nusantara (MTN) Jalan Tol JORR II Serpong-Kunciran kepemilikan saham sebesar 60%.
e. PT Marga Nujyasumo Agung (MNA) Jalan Tol Surabaya-Mojokerto kepemilikan saham sebesar 55% .
f. PT Marga Lingkar Jakarta (MLJ) Jalan Tol JORR W2 Utara kepemilikan saham sebesar 65% .
g. PT Trans Marga Jatim Pasuruan (TMJP) Jalan Tol Gempol-Pasuruan kepemilikan saham sebesar 80% .
h. PT Marga Kunciran Cengkareng (MKC) Jalan Tol JORR II Kunciran-Cengkareng kepemilikan saham sebesar 75% .
i. PT Margabumi Adhikaraya (MBAR) JAlan Tol Gempol-Pandaan dengan kepemilikan saham sebesar 52% .
j. PT Jasamarga Bali Tol Jalan Tol Benoa Bali dengan kepemilikan saham sebesar 60% .

Bidang Usaha Non Tol

Selain dari menambah jalan tol, Perseroan mengembangkan usaha lain dengan mengkapitalisasi berbagai aset-aset yang dimiliki perusahaan diantaranya adalah:

1. Penyewaan lahan dan Utilitas, saat ini Jasa Marga tengah menggarap jalur Serat Optik dari Bandung hingga Jakarta
2. pengembangan rest area, serta properti. Sampai akhir 2009, Perseroan telah membangun enam tempat istirahat (rest area), empat di antaranya berada di ruas Jakarta-Cikampek, satu di Bandung dan satu di Tangerang. Rencananya 2010 ini Perseroan akan membangun 14 tempat istirahat lagi di lokasi berbeda.
3. Pemasangan iklan,

4. Berbagai Jasa termasuk Jasa pengoperasian jalan tol pihak lain. Termasuk mengelola Jembatan Tol Suramadu yang menjadi kebanggan nasional. Selain itu Jasa Marga melalui anak perusahaan Sarana Marga Bhakti Utama telah melebarkan sayap ke berbagai bidang Jasa Lainnya seperti trnasportasi, pembangunan dan pemeliharaan jalan umum.Tahun 2009 lalu usaha lain-lain ini menyumbang pendapatan sebesar Rp 42,01 miliar, naik dari Rp 31,76 milar pada tahun sebelumnya.

Khusus pengelolaan Jembatan Suramadu, Terpilihnya Jasa Marga sebagai pengelola Jembatan sepanjang 5,4 km tersebut, menambah nilai tersendiri, yaitu kepercayaan dari Pemerintah terhadap pengalaman dan kemampuan yang dimiliki oleh perusahaan.

2.1.7 Tata Nilai

Tata Nilai merupakan nilai-nilai yang telah ada dalam setiap Insan Jasa Marga. Tata nilai ini merupakan perwujudan dari sikap dan perilaku seluruh karyawan Jasa Marga yang dilaksanakan untuk mendukung pencapaian tujuan perusahaan secara baik dan benar. Tata Nilai tersebut adalah:

1. Integritas
    a. Bekerja hanya untuk kepentingan Perusahaan Tidak pernah menyalahgunakan wewenang untuk kepentingan lain diluar kepentingan perusahaan
    b. Bertanggungjawab dan senantiasa dapat menjelaskan keputusan dan langkah-langkah yang diambil dalam pekerjaan
    c. Senantiasa menggunakan etika dalam bekerja
    d. Senantiasa menjadi panutan bagi lingkungannya
2. Mencintai Pekerjaan (*Passion*)
    a. Semangat dan keinginan yang kuat untuk senantiasa berbuat yang terbaik di bidangnya
    b. Menyenangi tugasnya dan selalu berpikir positif dalam bekerja
    c. Bangga terhadap Perusahaan sebagai wujud dari kebanggan pada Bangsa dan Negara
    d. Senantiasa menghasilkan kualitas pekerjaan yang terbaik

3. Senang Belajar untuk Kemajuan (*Learning*)
    a. Selalu ingin mengetahui dan belajar hal-hal baru untuk kemajuan perusahaan
    b. Melihat jauh kedepan dan senantiasa berusaha untuk membawa Perusahaan ke tingkat yang lebih tinggi.
    c. Berani mencoba hal-hal baru dengan niat semata-mata untuk memperbaiki kualitas proses dan produk Perusahaan.
4. Membangun Kepercayaan (*Trust*)
    a. Percaya pada niat baik
    b. Senantiasa membangun kepercayaan (trust) diantara seluruh jajaran Perusahaan
    c. Tidak terkotak-kotak, selalu saling membantu untuk kepentingan perusahaan semata

2.1.8 Kebijakan Mutu

Sebagai penyelenggara jasa jalan tol di Indonesia, Jasa Marga selalu berupaya meningkatkan pelayanan untuk mencapai sasaran mutu Lancar, Aman dan Nyaman. Dalam rangka mencapai sasaran tersebut, Jasa Marga menerapkan sistem mutu sesuai dengan standar ISO 9001:2000 dan untuk mendukung komitmen tersebut Jasa Marga menetapkan kebijakan mutu sebagai berikut:

1. Mengusahakan jasa pelayanan yang bermutu tinggi untuk memenuhi kelancaran, keamanan dan kenyamanan pelanggan.
2. Mendorong seluruh karyawan untuk selalu meningkatkan keterampilan dan keahlian serta selalu bertanggung jawab dan tertib dalam menjalankan tugas melayani pelanggan.
3. Terus menerus menyempurnakan sistem dan lingkungan kerja ke arah yang lebih efektif dan efisien untuk mendukung tercapainya mutu pelayanan

2.1.9 Pengukuran Tingkat Kepadatan Lalu Lintas

Berdasarkan hasil wawancara langsung dengan Ir.Darwan Edison selaku Kepala Bagian Pengumpulan Tol PT.Jasa Marga (Persero) Cabang Purbaleunyi, bahwa untuk pengukuran tingkat kepadatan lalu lintas di jalur tol belum ada

standar operasional (SOP) yang tertulis. Untuk pengukuran tingkat kepadatan lalu lintas dihitung berdasarkan pemantauan langsung dengan menghitung jumlah lalu lintas kendaraan tiap jam yaitu 3500 kendaraan per jam. Tingkat kepadatan lalu lintas dibagi kedalam 4 tingkatan yaitu sebagai berikut:

1. Sepi (jumlah kendaraan >=0 sampai <=1400 kendaraan per jam).
2. Sedang (jumlah kendaraan >1400 sampai <=2275 kendaraan per jam).
3. Padat(jumlah kendaraan >2275 sampai <=2975 kendaraan per jam).
4. Macet(jumlah kendaraan lebih dari 2975 kendaraan per jam).

2.2 Landasan Teori

Landasan teori berisi teori-teori tentang ilmu yang mendasari tentang bidang yang sedang diteliti. Penyajian teori dalam landasan teori bersumber dari buku, artikel dan modul.

2.2.1 Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi.

2.2.1.1 Konsep Dasar Sistem

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atauhimpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu:

a. Setiap sistem terdiri dari unsur-unsur.
b. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
c. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
d. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

2.2.1.2 Konsep Dasar Informasi

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Informasi merupakan data yang telah diklasifikasikanatau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

2.2.1.3 Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

2.2.1.4 Komponen dan Elemen Sistem Informasi

A. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan(building block), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen hardware, komponen software, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen input

   Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

   Komponen ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen output

   Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

   Teknologi merupakan “tool box” dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen hardware

   Hardware berperan penting sebagai suatu penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen software

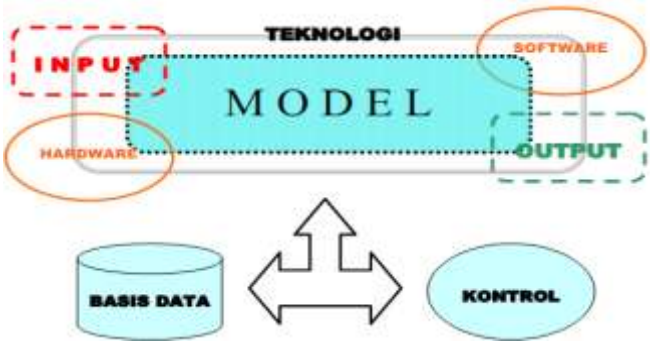
   Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen basis data

   Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS(*Database Management System*).

8. Komponen kontrol

   Banyak hal yang dapat merusak informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.



Gambar II.3 Interaksi Sistem Informasi

B. Elemen sistem informasi

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

1. Orang

   Orang atau personil yang dimaksudkan yaitu operator komputer, analis sistem, programmer, personil data entry, dan manajer sistem informasi.

2. Prosedur

   Prosedur merupakan elemen fisik. Hal ini disebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

3. Perangkat keras

   Perangkat keras bagi suatu sistem informasi terdiri atas komputer(pusat pengolah, unit masukan/keluaran), peralatan penyiapan data dan terminal masukan/keluaran.

4. Perangkat lunak

Perangkat lunak dapat dibagi dalam 3 jenis utama:

a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

5. Basis data

   File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti diskette, hard disk, magnetic tape dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film dan lain sebagainya.

6. Jaringan komputer

   Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

7. Komunikasi data

   Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain



Gambar II.4 Hubungan Elemen Sistem Informasi

2.2.1.5 Arsitektur dan Klasifikasi Sistem Informasi

A. Arsitektur Sistem Informasi

Sistem informasi dapat dibentuk sesuai kebutuhan organisasi masing-masing. Oleh karena itu, untuk dapat menerapkan sistem yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan dan evaluasi sesuai keinginan masing-masing organisasi. Guna dari sistem yang efektif dan efisien tidak lain untuk mendapatkan keunggulan dalam berkompetisi. Semua orang dapat menggunakan sistem informasi dalam organisasi, tetapi faktor efisiensi setiap sistem adalah berbeda.

Perlu diketahui, perubahan sistem, baik besar maupun kecil, selalu akan melalui tingkatan-tingkatan sebagai berikut:

Tingkat I : Ide, mengetahui perlu adanya perubahan.
Tingkat II : Design, merancang cara pemecahannya.
Tingkat III : Pelaksanaan, menerapkan design kedalam sistem.
Tingkat IV : Kontrol, memeriksa tingkat pelaksanaan dijalankan sesuai dengan design.
Tingkat V : Evaluasi, memeriksa apakah perubahan yang terjadi sesuai dengan tujuan semuala.
Tingkat VI : Tindak lanjut, melaksanakan perubahan sesuai dengan hasil evaluasi yang ada.

Adapun tingkatan yang menjadi kunci yang digunakan untuk memecahkan bagian masalah baik itu secara menyeluruh maupun per bagian, yaitu:



Gambar II.5 Tingkatan kunci untuk memecahkan masalah

B. Klasifikasi Sistem Informasi

Sistem informasi merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya:

a. Sistem abstrak atau sistem fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan.

b. Sistem alamiah dan sistem buatan manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi.

   Sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut human machine system. Sistem informasi berbasis internet merupakan contoh human machine system karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem deterministik dan sistem probabilistik

   Sistem deteministik adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi.

   Sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

d. Sistem terbuka dan sistem tertutup

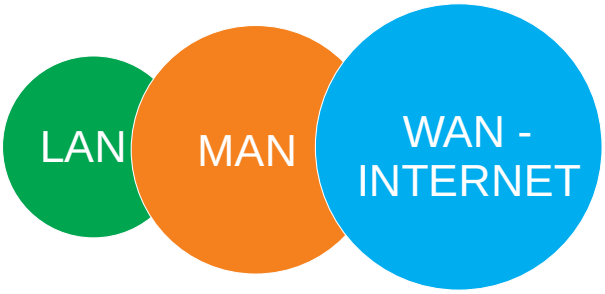
   Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya.

   Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

   Sistem tertutup adalah sistem yang tidak terhubung dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar.

2.2.2 Jaringan Komputer

Saat ini, internet dan world wide web (www) sangat populer di seluruh dunia. Banyak masyarakat yang membutuhkan aplikasi yang berbasis internet, seperti e-mail dan akses web melalui internet. Sehingga makin banyak aplikasi bisnis yang berkembang berjalan di atas internet.[2]



Gambar II.6 Jaringan Komputer

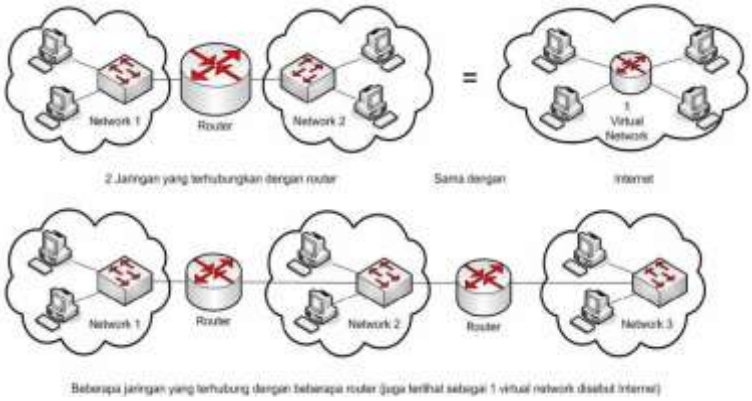
2.2.2.1 Model Arsitektur TCP/IP

TCP/IP(*Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol. Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (software) di sistem operasi.

TCP/IP merngimplemenasikan arsitektur berlapis yang terdiri atas empat lapis. Empat lapis ini, dapat dipetakan (meski tidak secara langsung) terhadap model referensi OSI. Empat lapis ini, kadang-kadang disebut sebagai DARPA Model, Internet Model, atau DoD Model, mengingat TCP/IP merupakan protokol yang awalnya dikembangkan dari proyek ARPANET yang dimulai oleh Departemen Pertahanan Amerika Serikat.

2.2.2.2 Internetworking

Tujuan dari TCP/IP adalah untuk membangun suatu koneksi antar jaringan (*network*), dimana biasa disebut internetwork, atau internet, yang menyediakan pelayanan komunikasi antar jaringan yang memiliki bentuk fisik yang beragam. Tujuan yang jelas adalah menghubungkan komputer(*hosts*) pada jaringan yang berbeda, atau mungkin terpisahkan secara geografis pada area yang luas.



Gambar II.7 Internetworking

Internet dapat digolongkan menjadi beberapa group jaringan, antara lain:

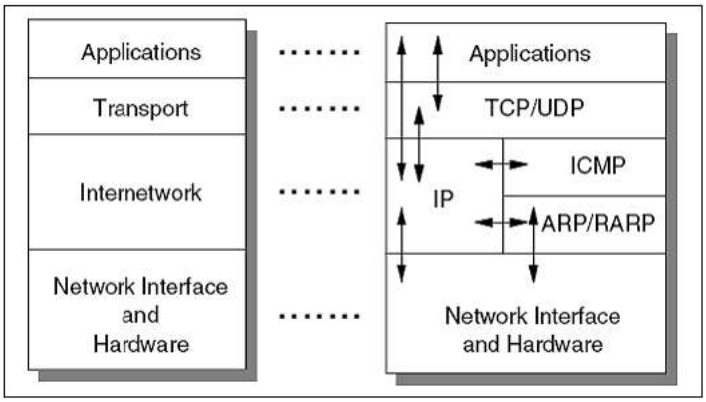
1. Backbone: Jaringan besar yang menghubungkan antar jaringan lainnya. Contoh: NSFNET yang merupakan jaringan backbone dunia di Amerika, EBONE yang merupakan jaringan backbone di Eropa, dan lainnya.
2. Jaringan regional, contoh: jaringan antar kampus.
3. Jaringanyangbersifatkomersialdimanamenyediakankoneksimenujubackboneke pada pelanggannya.
4. Jaringan lokal, contoh: jaringan dalam sebuah kampus.

Aspek lain yang penting dari TCP/IP adalah membentuk suatu standarisasi dalam komunikasi. Tiap-tiap bentuk fisik suatu jaringan memiliki teknologi yang berbeda-beda, sehingga diperlukan pemrograman atau fungsi khusus untuk digunakan dalam komunikasi. TCP/IP memberikan fasilitas khusus yang bekerja diatas pemrograman atau fungsi khusus tersebut dari masing-masing fisik jaringan. Sehingga bentuk arsitektur dari fisik jaringan akan tersamarkan dari pengguna dan pembuat aplikasi jaringan. Dengan TCP/IP, pengguna tidak perlu lagi memikirkan bentuk fisik jaringan untuk melakukan sebuah komunikasi.Untuk dapat berkomunikasi antar 2 jaringan, diperlukan komputer yang terhubung dalam suatu perangkat yang dapat meneruskan suatu paket data dari jaringan yang satu ke jaringan yang lain. Perangkat tersebut disebut Router. Selain itu router juga digunakan sebagai pengarah jalur (routing). Untuk dapat mengidentifikasikan komputer(*host*) diperlukan sebuah alamat, disebut alamat IP(*IP address*). Apabila

sebuah host memiliki beberapa perangkat jaringan (interface), seperti router, maka setiap interface harus memiliki sebuah IP address yang unik. IP address terdiri dari 2 bagian, yaitu : **IP address = <nomer jaringan><nomer host>**

2.2.2.3 Lapisan(layer) pada Protokol TCP/IP

Seperti pada perangkat lunak, TCP/IP dibentuk dalam beberapa lapisan(*layer*). Dengan dibentuk dalam layer, akan mempermudah untuk pengembangan dan pengimplementasian. Antar layer dapat berkomunikasi ke atas maupun ke bawah dengan suatu penghubung interface. Tiap-tiap layer memiliki fungsi dan kegunaan yang berbeda dan saling mendukung layer diatasnya. Pada protokol TCP/IP dibagi menjadi 4 layer, tampak pada Gambarberikut.



Gambar II.8 Protokol TCP/IP

Keterangan:

1. Layer Applications

   Layer aplikasi digunakan pada program untuk berkomunikasi menggunakan TCP/IP. Contoh aplikasi antaralain Telnet dan File Transfer Protocol (FTP). Interface yang digunakan untuk saling berkomunikasi adalah nomor port dan socket.

2. Layer Transport

   Layer transport memberikan fungsi pengiriman data secara end-to-end ke sisi remote. Aplikasi yang beragam dapat melakukan komunikasi secara serentak(*simulaneously*). Protokol pada layer transport yang paling sering digunakan adalah Transmission Control Protocol(TCP), dimana memberikan fungsi pengiriman data secara connection-oriented, pencegahan duplikasi data, congestion control dan flow control. Protokol lainnya adalah User Datagram

Protocol(UDP), dimana memberikan fungsi pengiriman connectionless, jalur yang tidak reliabel. UDP banyak digunakan pada aplikasi yang membutuhkan kecepatan tinggi dan dapat mentoleransi terhadap kerusakan data.

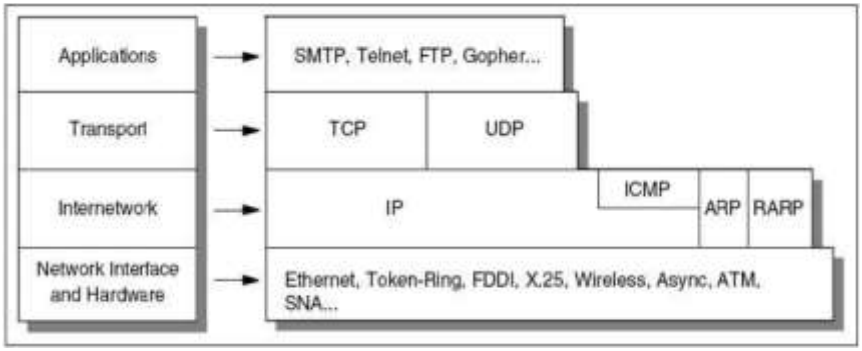
3. Layer Internetwork

   Layer Internetwork biasa disebut juga layer internet atau layer network, dimana memberikan "vitual network" pada internet. Internet Protocol (IP) adalah protokol yang paling penting. IP memberikan fungsi routing pada jaringan dalam pengiriman data. Protokol lainnya antaralain: IP, ICMP, IGMP, ARP, RARP Layer

4. Network Interface

   Layer network interface disebut juga layer link atau layer datalink, yang merupakanperangkatkeraspadajaringan.Contoh:IEEE802.2,X.25, ATM, FDDI, dan SNA.

   Secara detail dapat digambarkan pada Gambarberikut.



Gambar II.9 Detail dari model Arsitektur Protokol TCP/IP

Level tertinggi pada layer TCP/IP adalah aplikasi. Dimana layer ini melakukan komunikasi sehingga dapat berinteraksi dengan pengguna.

Karakteristik dari protokol aplikasi antara lain:

1. Merupakan program aplikasi yang dibuat oleh pengguna, atau aplikasi yang merupakan standar dari produk TCP/IP. Contoh aplikasi yang merupakan produk dari TCP/IP antara lain :
   a. TELNET, terminal interaktif untuk mengakses suatu remote pada internet.
   b. HTTP (Hyper Text Transfer Protocol), pengiriman file hypertext ukuran besar.

c. FTP (File Transfer Protocol), transfer file berkecepatan tinggi antar komputer
d. SMTP (Simple Mail Transfer Protocol), sistem bersurat di internet
e. dll

2. Menggunakan mekanisme TCP atau UDP.
3. Menggunakan model interaksi client/server.

2.2.2.4 Topologi Jaringan Komputer

Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Pola ini sangat erat kaitannya dengan metode access dan media pengiriman yang digunakan. Topologi yang ada sangatlah tergantung dengan letak geofrapis dari masing-masing terminal, kualitas kontrol yang dibutuhkan dalam komunikasi ataupun penyampaian pesan, serta kecepatan dari pengiriman data. Dalam definisi topologi terbagi menjadi dua, yaitu topologi fisik (physical topology) yang menunjukan posisi pemasangan kabel secara fisik dan topologi logik (logical topology) yang menunjukan bagaimana suatu media diakses oleh host. Adapun topologi fisik yang umum digunakan dalam membangun sebuah jaringan adalah :

1. Point to Point (Titik ke-Titik).

   Jaringan kerja titik ketitik merupakan jaringan kerja yang paling sederhana tetapi dapat digunakan secara luas. Begitu sederhananya jaringan ini, sehingga seringkali tidak dianggap sebagai suatu jaringan tetapi hanya merupakan komunikasi biasa. Dalam hal ini, kedua simpul mempunyai kedudukan yang setingkat, sehingga simpul manapun dapat memulai dan mengendalikan hubungan dalam jaringan tersebut. Data dikirim dari satu simpul langsung kesimpul lainnya sebagai penerima, misalnya antara terminal dengan CPU.

2. Star Network (Jaringan Bintang).

   Dalam konfigurasi bintang, beberapa peralatan yang ada akan dihubungkan kedalam satu pusat komputer. Kontrol yang ada akan dipusatkan pada satu titik, seperti misalnya mengatur beban kerja serta pengaturan sumber daya yang ada. Semua link harus berhubungan dengan pusat apabila ingin menyalurkan data kesimpul lainnya yang dituju. Dalam hal ini, bila pusat

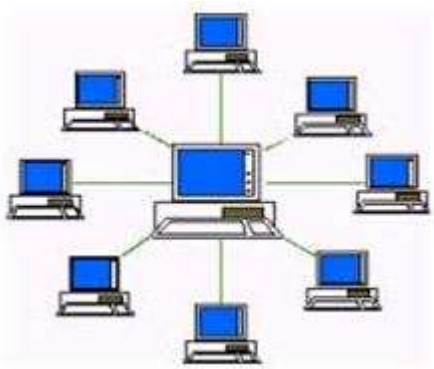
mengalami gangguan, maka semua terminal juga akan terganggu. Model jaringan bintang ini relatif sangat sederhana, sehingga banyak digunakan oleh pihak per-bank-kan yang biasanya mempunyai banyak kantor cabang yang tersebar diberbagai lokasi. Dengan adanya konfigurasi bintang ini, maka segala macam kegiatan yang ada di-kantor cabang dapatlah dikontrol dan dikoordinasikan dengan baik. Disamping itu, dunia pendidikan juga banyak memanfaatkan jaringan bintang ini guna mengontrol kegiatan anak didik mereka.

Kelebihan

a. Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan station yang terpaut.
b. Tingkat keamanan termasuk tinggi.
c. Tahan terhadap lalu lintas jaringan yang sibuk.
d. Penambahan dan pengurangan station dapat dilakukan dengan mudah.

Kekurangan

a. Jika node tengah mengalami kerusakan, maka maka seluruh jaringan akan terhenti.
b. Penanganan: Perlunya disiapkan node tengah cadangan.

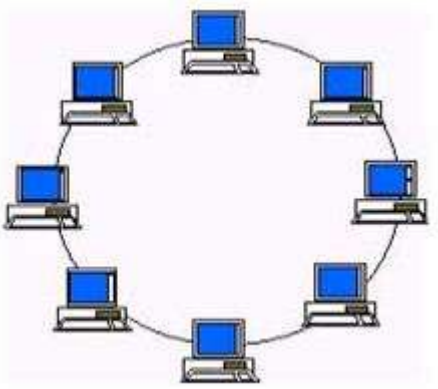


Gambar II.10 Topologi jaringan bintang

3. Ring Networks (Jaringan Cincin)

Pada jaringan ini terdapat beberapa peralatan saling dihubungkan satu dengan lainnya dan pada akhirnya akan membentuk bagan seperti halnya sebuah cincin. Jaringan cincin tidak memiliki suatu titik yang bertindak sebagai pusat ataupun pengatur lalu lintas data, semua simpul mempunyai tingkatan yang

sama. Data yang dikirim akan berjalan melewati beberapa simpul sehingga sampai pada simpul yang dituju. Dalam menyampaikan data, jaringan bisa bergerak dalam satu ataupun dua arah. Walaupun demikian, data yang ada tetap bergerak satu arah dalam satu saat. Pertama, pesan yang ada akan disampaikan dari titik ketitik lainnya dalam satu arah. Apabila ditemui kegagalan, misalnya terdapat kerusakan pada peralatan yang ada, maka data yang ada akan dikirim dengan cara kedua, yaitu pesan kemudian ditransmisikan dalam arah yang berlawanan, dan pada akhirnya bisa berakhir pada tempat yang dituju. Konfigurasi semacam ini relative lebih mahal apabila dibanding dengan konfigurasi jaringan bintang. Hal ini disebabkan, setiap simpul yang ada akan bertindak sebagai komputer yang akan mengatasi setiap aplikasi yang dihadapinya, serta harus mampu membagi sumber daya yang dimilikinya pada jaringan yang ada. Disamping itu, sistem ini lebih sesuai digunakan untuk sistem yang tidak terpusat (decentralized-system), dimana tidak diperlukan adanya suatu prioritas tertentu.

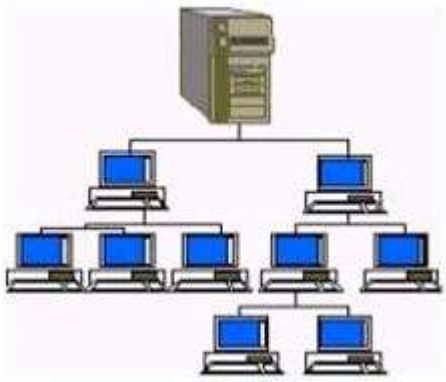


Gambar II.11 Topologi jaringan cincin

4. Tree Network (Jaringan Pohon)

   Pada jaringan pohon, terdapat beberapa tingkatan simpul (node). Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Misalnya untuk bergerak dari komputer dengan node-3 kekomputer node-7 seperti halnya pada gambar, data yang ada harus melewati node-3, 5 dan node-6 sebelum berakhir pada node-7. Keungguluan jaringan model pohon seperti ini adalah, dapat terbentuknya suatu kelompok yang dibutuhkan pada setiap saat. Sebagai contoh, perusahaan dapat membentuk kelompok yang

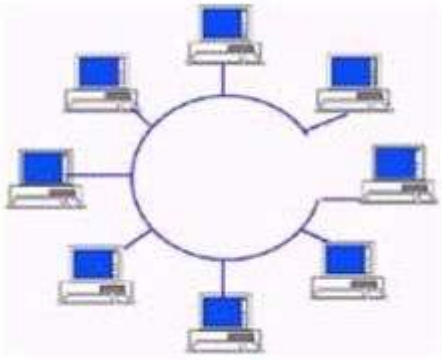
terdiri atas terminal pembukuan, serta pada kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah, apabila simpul yang lebih tinggi kemudian tidak berfungsi, maka kelompok lainnya yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif menjadi lambat



Gambar II.12 Topologi jaringan pohon

5. Bus Network

   Konfigurasi lainnya dikenal dengan istilah bus-network, yang cocok digunakan untuk daerah yang tidak terlalu luas. Setiap komputer (setiap simpul) akan dihubungkan dengan sebuah kabel komunikasi melalui sebuah interface. Setiap komputer dapat berkomunikasi langsung dengan komputer ataupun peralatan lainnya yang terdapat didalam network, dengan kata lain, semua simpul mempunyai kedudukan yang sama. Dalam hal ini, jaringan tidak tergantung kepada komputer yang ada dipusat, sehingga bila salah satu peralatan atau salah satu simpul mengalami kerusakan, sistem tetap dapat beroperasi. Setiap simpul yang ada memiliki address atau alam sendiri. Sehingga untuk meng-access data dari salah satu simpul, user atau pemakai cukup menyebutkan alamat dari simpul yang dimaksud. Keunggulan topologi Bus adalah pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.



Gambar II.13 Topologi jaringan bus

6. Plex Network (Jaringan Kombinasi)

   Merupakan jaringan yang benar-benar interaktif, dimana setiap simpul mempunyai kemampuan untuk meng-access secara langsung tidak hanya terhadap komputer, tetapi juga dengan peralatan ataupun simpul yang lain. Secara umum, jaringan ini mempunyai bentuk mirip dengan jaringan bintang. Organisasi data yang ada menggunakan de-sentralisasi, sedang untuk melakukan perawatan, digunakan fasilitas sentralisasi.



Gambar II.14 Topologi jaringan kombinasi

Topologi Logik pada umumnya terbagi mejadi dua tipe, yaitu :

a. Topologi Broadcast

   Secara sederhana dapat digambarkan yaitu suatu host yang mengirimkan data kepada seluruh host lain pada media jaringan.

b. Topologi Token Passing

Mengatur pengiriman data pada host melalui media dengan menggunakan token yang secara teratur berputar pada seluruh host. Host hanya dapat mengirimkan data hanya jika host tersebut memiliki token. Dengan token ini, collision dapat dicegah.

Faktor – faktor yang perlu mendapat pertimbangan untuk pemilihan topologi adalah sebagai berikut :

a. Biaya

   Sistem apa yang paling efisien yang dibutuhkan dalam organisasi.

b. Kecepatan

   Sampai sejauh mana kecepatan yang dibutuhkan dalam sistem.

c. Lingkungan

   Misalnya listrik atau faktor – faktor lingkungan yang lain, yang berpengaruh pada jenis perangkat keras yang digunakan.

d. Ukuran

   Sampai seberapa besar ukuran jaringan. Apakah jaringan memerlukan file server atau sejumlah server khusus.

e. Konektivitas

   Apakahpemakaiyanglainyangmenggunakankomputerlaptopperlu mengakses jaringan dari berbagai lokasi.

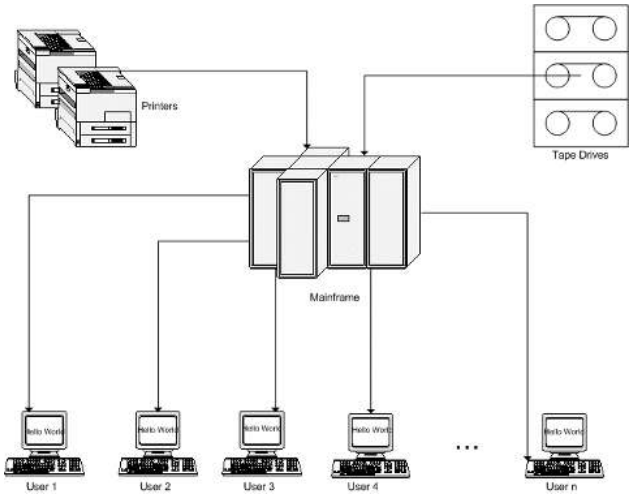
2.2.3 Arsitektur Client Server

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu sistem ditempatkan dan bagaimana mereka berkomunikasi. Arsitektur terdistribusi – sebuah istilah yang relatif baru untuk menjelaskan arsitektur aplikasi – berarti bahwa pemrosesan dari suatu aplikasi terjadi pada lebih dari satu mesin.

Terdapat beberapa macam arsitektur aplikasi, yaitu :

1. Standalone (one-tier)

   Pada arsitektur ini semua pemrosesan dilakukan pada mainframe. Kode aplikasi, data dan semua komponen sistem ditempatkan dan dijalankan pada host. Seperti terlihat pada gambar 2.13.

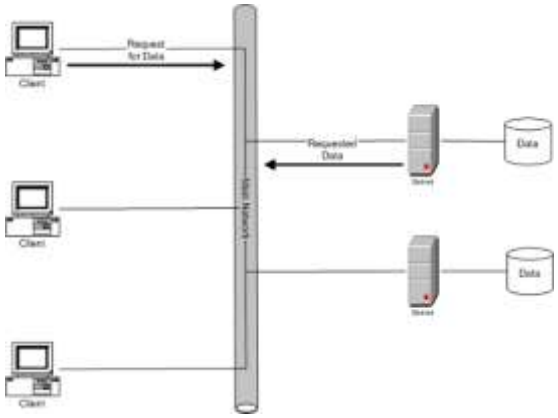


Gambar II.15 Arsitektur *Standalone (one-tier)*

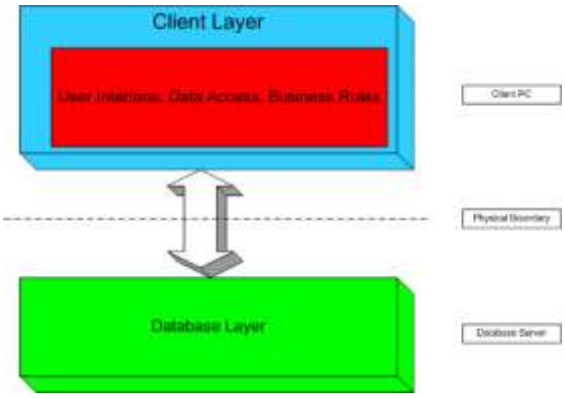
Walaupun komputer client dipakai untuk mengakses mainframe, tidak ada pemrosesan yang terjadi pada mesin ini, dan karena mereka "dump-client" atau "dump-terminal". Tipe model ini, dimana semua pemrosesan terjadi secara terpusat, dikenal sebagai berbasis-host. Sekilas dapat dilihat kesalahan pada model ini. Ada dua masalah pada komputasi berbasis host: Pertama, semua pemrosesan terjadi pada sebuah mesin tunggal, sehingga semakin banyak user yang mengakses host, semakin kewalahan jadinya. Jika sebuah perusahaan memiliki beberapa kantor pusat, user yang dapat mengakses mainframe adalah yang berlokasi pada tempat itu, membiarkan kantor lain tanpa akses ke aplikasi yang ada.

2. Client/Server (two-tier)

   Dalam model client/server, pemrosesan pada sebuah aplikasi terjadi pada client dan server. Client/server adalah tipikal sebuah aplikasi two-tier dengan banyak client dan sebuah server yang dihubungkan melalui sebuah jaringan, seperti terlihat dalam gambar 2.14. Aplikasi ditempatkan pada komputer client dan mesin database dijalankan pada server jarak-jauh. Aplikasi client mengeluarkan permintaan ke database yang mengirimkan kembali data ke client-nya.



Gambar II.16 ***Client Server Physical Model***



Gambar II.17 Arsitektur ***Client/Server (two-tier) - Logical View***

Dalam client/server, client-client yang cerdas bertanggung jawab untuk bagian dari aplikasi yang berinteraksi dengan user, termasuk logika bisnis dan komunikasi dengan server database. Tipe-tipe tugas yang terjadi pada client adalah :

a. Antarmuka pengguna
b. Interaksi database
c. Pengambilan dan modifikasi data
d. Sejumlah aturan bisnis
e. Penanganan kesalahan

Server database berisi mesin database, termasuk tabel, prosedur tersimpan, dan trigger (yang juga berisi aturan bisnis). Dalam sistem client/server, sebagian besar logika bisnis biasanya diterapkan dalam database. Server database manangani :

a. Manajemen data
b. Keamanan
c. Query, trigger, prosedur tersimpan
d. Penangan kesalahan

Arsitektur client/server merupakan sebuah langkah maju karena mengurangi beban pemrosesan dari komputer sentral ke komputer client. Ini berarti semakin banyak user bertambah pada aplikasi client/server, kinerja server file tidak akan menurun dengan cepat. Dengan client/server user dair berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal. Namun masih terdapat kelemahan pada model ini. Selain menjalankan tugas-tugas tertentu, kinerja dan skalabilitas merupakan tujuan nyata dari sebagian besar aplikasi. Model client/server memiliki sejumlah keterbatasan :

a. Kurangnya skalabilitas
b. Koneksi database dijaga
c. Tidak ada keterbaharuan kode
d. Tidak ada tingkat menengah untuk menangani keamanan dan transaksi

Aplikasi-aplikasi berbasis client/server memiliki kekurangan pada skalabilitas. Skalabilitas adalah seberapa besar aplikasi bisa menangani suatu kebutuhan yang meningkat – misalnya, 50 user tambahan yang mengakses aplikasi tersebut. Walaupun model client/server lebih terukur daripada model berbasis host, masih banyak pemrosesan yang terjadi pada server. Dalam model client/server semakin banyak client yang menggunakan suatu aplikasi, semakin banyak beban pada server.

Koneksi database harus dijaga untuk masing-masing client. Koneksi menghabiskan sumber daya server yang berharga dan masing-masing client tambahan diterjemahkan ke dalam satu atau beberapa koneksi. Logika kode tidak bisa didaur ulang karena kode aplikasi ada dalam sebuah pelaksanaan execuTabel monolitik pada client. Ini juga menjadikan modifikasi pada kode sumber sulit. Penyusunan ulang perubahan itu ke semua komputer client juga membuat sakit kepala.

Keamanan dan transaksi juga harus dikodekan sebagai pengganti penanganan oleh COM+/MTS. Bukan berarti model client/server bukanlah merupakan model yang layak bagi aplikasi-aplikasi. Banyak aplikasi yang lebih kecil dengan jumlah user terbatas bekerja sempurna dengan model ini. Kemudahan pengembangan aplikasi client/server turut menjadikannya sebuah solusi menarik bagi perusahaan.

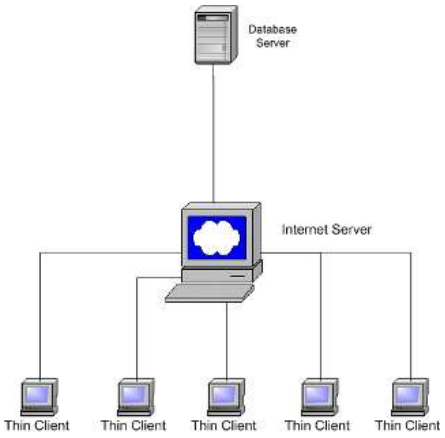
Pengembangan umumnya jauh lebih cepat dengan tipe sistem ini. Siklus pengembangan yang lebih cepat ini tidak hanya menjadikan aplikasi meningkat dan berjalan dengan cepat namun juga lebih hemat biaya.

3. Three-Tier / Multi-Tier

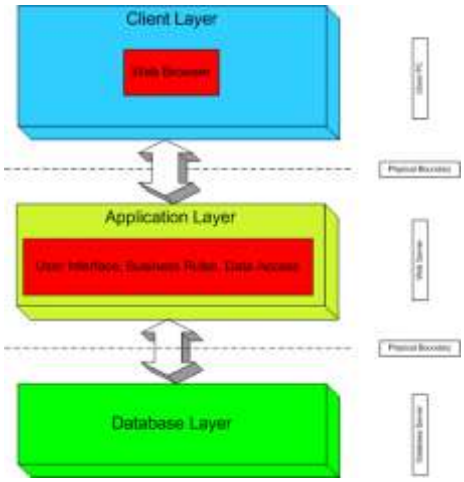
Model three-tier atau multi-tier dikembangkan untuk menjawab keterbatasan pada arsitektur client/server. Dalam model ini, pemrosesan disebarkan di dalam tiga lapisan (atau lebih jika diterapkan arsitektur multitier). Lapisan ketiga dalam arsitektur ini masing-masing menjumlahkan fungsionalitas khusus. Yaitu :

a. Layanan presentasi (tingkat client)
b. Layanan bisnis (tingkat menengah)
c. Layanan data (tingkat sumber data)

Layanan presentasi atau logika antarmuka pengguna ditempatkan pada mesin client. Logika bisnis dikeluarkan dari kode client dan ditempatkan dalam tingkat menengah. Lapisan layanan data berisi server database. Setiap tingkatan dalam model three-tier berada pada komputer tersendiri, seperti pada gambar 2.16.



Gambar II.18 Three-Tier / Multi-Tier



Gambar II.19 Three-Tier/Multi-Tier - Logical View

Konsep model three-tier adalah model yang membagi fungsionalitas ke dalam lapisan-lapisan, aplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

2.2.4 Basis Data

Basis data adalah sekumpulan data yang terintegrasi yang diorganisasi untuk memenuhi kebutuhan pemakai untuk keperluan organisasi[4].

DBMS (Data Base Management System) adalah perangkat lunak yang menangani semua pengaksesan ke data base.

Sistem Basis Data = DBMS + Basis data

Struktur File Database

1. Data adalah satu satuan informasi yang akan diolah, dimana sebelum diolah dikumpulkan di dalam suatu file database. Pengumpulan data dilakukan secara sistematis menurut struktur file database tersebut.
2. RECORD adalah data yang isinya merupakan satu kesatuan seperti Nama, Alamat, Nomor Telepon. Setiap keterangan yang mencakup Nama, Alamat dan Nomor Telepon dinamakan satu record. Dan setiap record diberi nomor urut yang disebut nomor record (Record Number). Ukuran suatu file database ditentukan oleh jumlah record yang tersimpan di dalamnya.

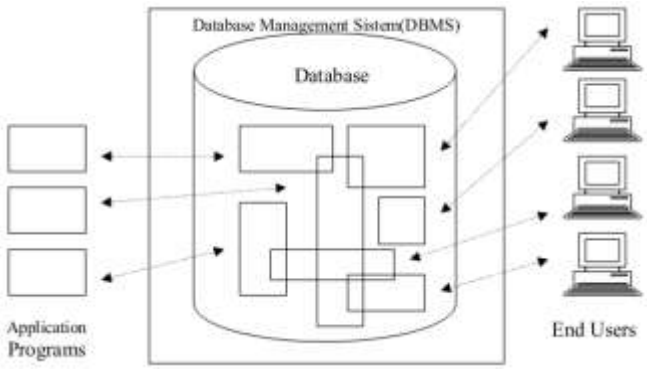
3. FIELD adalah sub bagian dari Record. Dari contoh isi record diatas maka terdiri dari 3 field, yaitu field Nama, field Alamat dan field Nomor Telepon.

Tabel II.1 Perbedaan dan Kelemahan File Manajemen

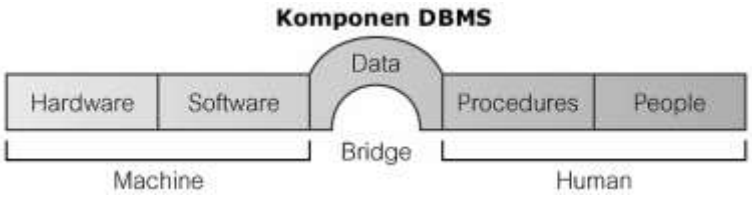
Perbedaan	
File manajemen tradisional	File manajemen data base
1. Program Oriented 2. Kaku 3. Kerangkapan data	1. Data Oriented 2. Luwes 3. Tidak terjadi kerangkapan data
Kelemahan	
File manajemen tradisional	File manajemen data base
1. Timbulnya data rangkap dan ketidak konsistenan data 2. Data tidak dapat digunakan secara bersama-sama 3. Kesukaran dalam mengakses data 4. Tidak fleksibel 5. Data tidak standar	1. Storage yang dibutuhkan besar 2. Dibutuhkan tenaga spesialis 3. Software mahal 4. Kerusakan pada data base dapat mempengaruhi departemen lain yang terkait kerangkapan data

Keuntungan file manajemen data base :

1. Tidak terjadi kerangkapan data
2. Data lebih konsisten
3. Data dapat digunakan bersama-sama
4. Data dapat distandarisasi
5. Keamanan data dapat terjamin
6. Integritas data terpelihara
7. Data independen



Gambar II.20 Sistem basis data



Gambar II.21 Komponen DBMS

Hardware

a. Meliputi PC sampai dengan jaringan komputer.
b. Tempat penyimpanan secondary (manegtic disk), I/O device ex : disk drives), device Controller, I/O Channels, dan lainnya.
c. Hardware processor dan main memory, digunakan untuk mendukung saat eksekusi system software database.

Software

a. DBMS, operating system, network software (jika diperlukan) dan program aplikasi pendukung lainnya.

Data

a. Data pada sebuah system database baik itu single-user system maupun multi-user system harus terintegrasi dan dapat digunakan bersama (*Integrated and Shared*).
b. Digunakan oleh organisasi dan deskripsi dari data disebut schema.

Procedures

a. Instrukti dan aturan yang harus disertakan dalam mendesain dan menggunakan database dan DBMS.

People

a. DA (*Data Administrator*), seseorang yang berwenang untuk membuat keputusan stategis dan kebijakan mengenai data yang ada
b. DBA (DataBase Administrator), menyediakan dukungan teknis untuk implementasi keputusan tersebut, dan bertanggung jawab atas keseluruhan kontrol system pada level teknis
c. Database Designer (Logical and Physical)

d. Application Programmers, bertanggungjawab untuk membuat aplikasi database dengan menggunakan bahasa pemrograman yang ada, seperti : C++, Java, dan lainnya.
e. End Users, Siapapun yang berinteraksi dengan sistem secara
f. online melalui workstation/terminal.

Kerugian DBMS

a. Rumit (Complexity)

   Karena penetapan fungsi dari DBMS yang baik, menyebabkan DBMS menjadi software yang cukup rumit. Seluruh user harus mengetahui fungsi-fungsi yang ada dengan baik, sehingga dapat memperoleh manfaatnya.

b. Ukuran (*Size*)

   Kerumitan dan banyaknya fungsi yang ada menyebabkan DBMS memerlukan banyak software pendukung yang mengakibatkan penambahan tempat penyimpanan dan memory.

c. Biaya DBMS (*Cost of DBMS*)
d. Biaya Tambahan Hardware (*Additional hardware costs*)
e. Biaya Konversi (*Cost of conversion*)
f. Performance

   Pada dasarnya DBMS dibuat untuk menyediakan banyak aplikasi, akibatnya mungkin beberapa aplikasi akan berjalan tidak seperti biasanya.

g. Higher impact of a failure

   Karena system yang terpusat, jika seluruh user dan aplikasi terakses dari DBMS maka kerusakan pada bagian manapun dari sistem, akan menyebabkan operasi terhenti.

Fungsi-fungsi DBMS

1. Data definition

   DBMS harus dapat mengolah pendefinisian data

2. Data manipulation

   DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data

3. Data security and integrity

DBMS harus dapat memeriksa keamanan dan integriti data yang didefinisikan oleh DBA

4. Data recovery and concurrency

   DBMS harus dapat menangani kegagalan pengaksesan database yang disebabkan oleh kesalahan system, kerusakan disk dan sebagainya. DBMS harus dapat memantau pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat bersamaan

5. Data dictionary

   Tempat penyimpanan informasi yang menggambarkan data dalam database. Data dictionary disebut juga metadata (data mengenai data) Berisi tentang:

   a. Nama-nama user yang mempunyai wewenang untuk penggunaan DBMS
   b. Nama-nama item data
   c. Jenis-jenis dan ukuran item data
   d. Batasan untuk masing-masing item data

6. Performance

   DBMS harus dapat menangani unjuk kerja dari semua fungsi se-efisien mungkin

Model Data

Kumpulan konsep-konsep yang terintegrasi untuk menggambarkan data, relationships antar data, dan batasan-batasan data dalam organisasi.

a. Data Model terdiri dari :
   a. Bagian struktural, berisikan sekumpulan aturan berdasarkan database yang dapat dibuat.
   b. Bagian manipulasi, mendefinisikan tipe operasi yang boleh dilakukan.
   c. Aturan-aturan Integritas.
b. Kegunaan untuk
   a. Merepresentasikan data kedalam bentuk yang lebih mudah untuk dipahami.
   b. Untuk menetapkan konsistensi dalam memandang, mengorganisir, menginterpretasikan dan memperlakukan database.
c. Jenis-jenis Model data :

a. Object-Based Data Models Entity-Relationship, Semantic, Functional, Object-Oriented.
b. Record-Based Data Models Relational Data Model, Network Data Model, Hierarchical Data Model.
c. Physical Data Models: Menerangkan bagaimana data disimpan dalam komputer, merepresentasikan informasi seperti ; struktur record, permintaan record, dan jalur akses
d. Conceptual Modelling
    1. Skema konseptual merupakan bagian utama dari system yang menampilkan view seluruh user.
    2. Merupakan representasi yang akurat dan lengkap dari kebutuhan data pada organisasi.
    3. Merupakan proses pembentukan suatu model informasi yang digunakan dalam organisasi yang terlepas dari detail implementasi.
    4. Hasilnya merupakan model data konseptual.

Basis data relasional (relational database) ialah model data yang berbentuk tabel dua dimensi yang terdiri atas baris dan kolom.

Keuntungan basis data relasional

1. bentuknya sederhana
2. mudah melakukan berbagai operasi data

Istilah dalam basis data relasional

1. relasi : sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris
2. atribut/field : kolom pada sebuah relasi
3. tupel/record : baris pada sebuah relasi
4. domain : kumpulan nilai yang valid untuk satu atau lebih atribut
5. degree (derajat) : jumlah atribut dalam sebuah relasi
6. cardinality : jumlah tupelo dalam sebuah relasi

Relational key :

1. Super key

Satu atau kumpulan atribut yang secara unik mengidetifikasi sebuah tupel didalam relasi.

2. Candidate key

   Atribut didalam relasi yang biasanya mempunyai nilai unik.

3. Primary key

   Candidate key yang dipilih untuk mengidentifikasikan tupel secara unik dalam relasi.

4. Alternate key

   Candidate key yang tidak terpilih sebagai primary key.

5. Foreign key

   Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain atribut tersebut hanya menjadi atribut biasa.

Relational integrity role :

1. Null

   Nilai suatu atribt yang tidak diketahui dan tidak cocok untuk tupel tersebut.

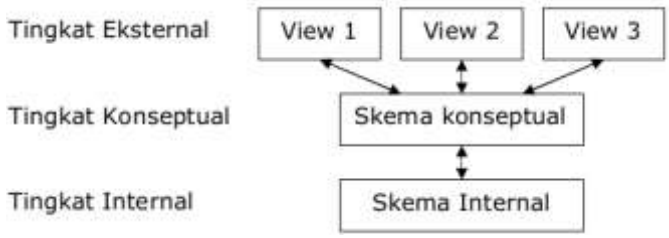
2. Entity Integrity

   Tidak ada satu komponen primary key yang bernilai null.

3. Referential Integrity

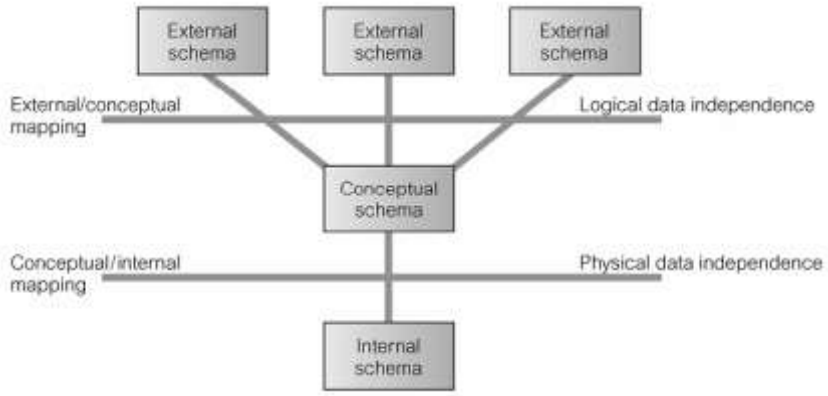
   Suatu domain dapat dipakai sebagai kunci utama bila merupakan atribut tunggal pada domain yang bersangkutan.

Lingkungan basis data



Gambar II.22 Lingkungan basis data1

Tingkat eksternal yaitu menerangkan view data base dari sekelompok pemakai. Tingkat konseptual yaitu menerangkan informasi database secara menyeluruh dengan menyembunyikan data secara fisik. Tingkat internal yaitu menerangkan struktur penyimpanan database secara fisik.



Gambar II.23 Lingkungan basis data2

Data independen : kapasitas untuk mengubah skema pada satu level sistem basis data tanpa mengubah skema pada level lain.

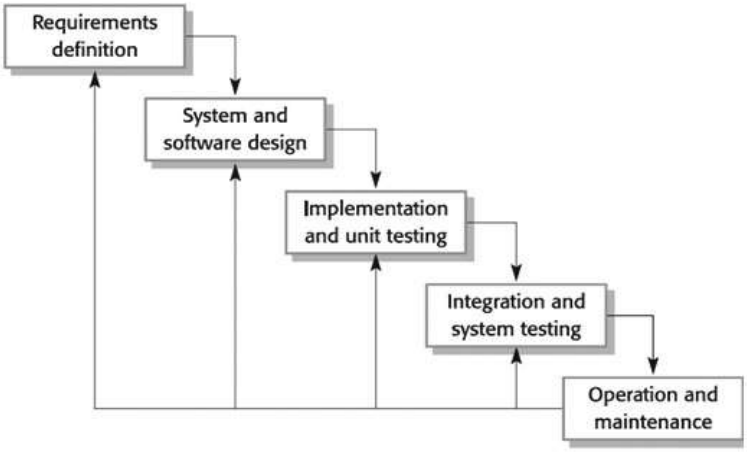
1. Physical Data Independent, perubahan pada skema internal tidak mempengaruhi skema lain.
2. Logical Data Independent, perubahan pada skema konseptual tidak mempengaruhi skema lain.

Mapping (Transformasi), proses pendefinisian informasi dari satu level ke level lainnya.

1. Konseptual /internal mapping, pendefinisian hubungan antara view konseptual dengan database dilevel internal.
2. Eksternal /konseptual mapping, pendefinisian hubungan antara view konseptual dengan database dilevel eksternal.

2.2.5 Konsep Pemodelan *Waterfall*

Waterfall model merupakan model lama namun sangat beralasan digunakan ketika kebutuhan dari sistem telah dipahami dengan baik. Gambaran dari konsep pemodelan *waterfall* dapat dilihat pada gambar II.24



Gambar II.24 Pemodelan *Waterfall*

1. ***Requirements analysis and definition:*** Mengumpulkan kebutuhan secara lengkap kemudian kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.
2. ***System and software design:*** Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
3. ***Implementation and unit testing:*** desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.
4. ***Integration and system testing:*** Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).
5. ***Operation and maintenance:*** mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesaisebelum mengerjakan fase berikutnya.

Masalah dengan *waterfall* :

1. Perubahan sulit dilakukan karena sifatnya yang kaku.
2. Karena sifat kakunya, model ini cocok ketika kebutuhan dikumpulkan secaralengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi pada kenyataannya jarang sekali konsumen/pengguna yang bisa memberikan

kebutuhan secara lengkap, perubahan kebutuhan adalah sesuatu yang wajar terjadi

2.2.6 PENGOLAHAN CITRA

Pengolahan citra merupakan kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia/mesin (komputer). masukannya adalah citra dan keluarannya juga citra tapi dengan kualitas lebih baik daripada citra masukan, misal citra warnanya kurang tajam, kabur (*blurring*), mengandung *noise* (misal bintik-bintik putih), sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi berkurang. Operasi-operasi pada pengolahan citra diterapkan pada citra bila[6] :

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra *(image enhancement).* Contoh : perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll
2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan *(image restoration)*. Contoh : penghilangan kesamaran (*debluring*) citra tampak kabur karena pengaturan fokus lensa tidak tepat / kamera goyang, penghilangan *noise*.
3. Elemen dalam citra perlu dikelompokkan, dicocokan atau diukur *(image segmentation).* Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek *(image analysis).* Proses segementasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh : pendeteksian tepi objek
5. Sebagian citra perlu digabung dengan bagian citra yang lain *(image reconstruction)*. Contoh : beberapa foto rontgen digunakan untuk membentuk ulang gambar organ tubuh
6. Citra perlu dimampatkan *(image compression)*

Contoh : suatu file citra berbentuk BMP berukuran 258 KB dimampatkan dengan metode JPEG menjadi berukuran 49 KB.

7. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang *(steganografi* dan *watermarking)*

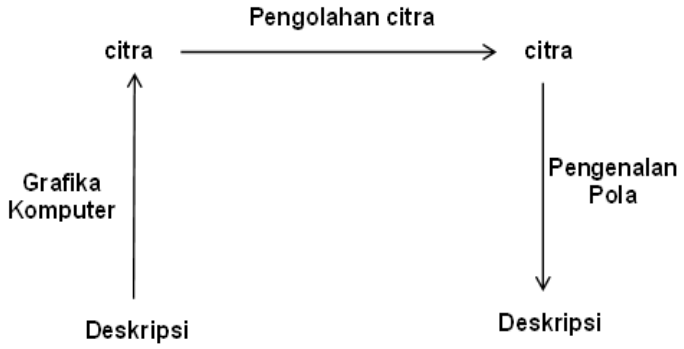
Beberapa jenis operasi pengolahan citra adalah sebagai berikut:

1. Modifikasi Kecemerlangan(*Brightness Modification*)
   Mengubah nilai keabuan/warna dari gelap menuju terang atau sebaliknya mengubah citra yang terlalu cemerlang/pucat menjadi gelap.
2. Peningkatan Kontras (*Contast Enhancement*)
   Dengan peningkatan kontras maka titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih cemerlang.
3. Negasi
   Operasi untuk mendapatkan citra negatif (*negative image*)
4. Pengabuan (*grayscale*)
   Merupakan proses konversi citra dengan warna sebenarnya (*true color*) menjadi citra keabuan (*grayscale*).
5. Pengambangan (*Thresholding*)
   Operasi pengambangan digunakan untuk mengubah citra dengan format skala keabuan, yang mempunyai kemungkinan nilai lebih dari 2 ke citra biner yang memiliki 2 buah nilai (yaitu 0 dan 1).
6. Pencerminan (*Flipping*)
   Pencerminan merupakan proses menggambar citra ke bentuk kebalikannya seperti ketika sedang bercermin.
7. Rotasi (*Rotating*)
   Rotasi yaitu proses memutar koordinat citra sesuai derajat yang ditentukan.
8. Pemotongan (*Cropping*)
   Memotong satu bagian dari citra sesuai kebutuhan.
9. Pengskalaan (*Scaling*)
   Mengubah ukuran citra menjadi lebih besar atau lebih kecil.
10. Deteksi Tepi (*Edge Detection*)

Deteksi tepi *(edge detection)* pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra

Didalam bidang komputer, ada 3 bidang studi yang berkaitan dengan citra, namun tujuan ketiganya berbeda, yaitu:

1. GrafikaKomputer(Computer Graphics)
2. PengolahanCitra (Image Processing)
3. PengenalanPola(Pattern Recognition/image interpretation)



Gambar II.25 Bidang Studi Yang Berkaitan Dengan Citra

COMPUTER VISION

Computer Vision mencoba meniru *Human Vision*

Computer Vision = proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti:

1. Akuisisi citra
2. Pengolahan citra
3. Klasifikasi
4. Pengenalan (Recognition)
5. Membuat Keputusan.

Vision = Geometry + Measurement + Interpretation

Proses-proses dalam *computer vision* dapat dibagi menjadi 3 aktivitas:

1. Memperoleh atau mengakuisisi citra digital
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra (operasi-operasi pengolahan citra)

3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya mengontrol peralatan, memantau proses manufaktur, memandu robot, dll.

Aplikasi pengolahan citra

1. Bidang Perdagangan
    a. Pembacaan *barcode*
    b. Pengenalan huruf/angka pada suatu formulir secara otomatis
2. Bidang Militer
    a. Mengenali sasaran peluru kendali melalui sensor visual
    b. Mengidentifikasi jenis pesawat musuh
3. Bidang Kedokteran
    a. Mammografi
    b. Rekontruksi foto janin hasil USG
4. Bidang Biologi
    a. Pengenalan jenis kromosom melalui citra mikroskopik
5. KomunikasiData
    a. Kompresi citra yang akan ditransmisikan
6. Hiburan
    a. Game
    b. Kompresi Video
7. Robotika
    a. Visualy-Guided autonomous navigation
8. Pemetaan
    a. Klasifikasi penggunaan tanah melalui foto udara/LANDSAT
9. Geologi
    a. Mengenali jenis batu-batuan melalui foto udara/LANDSAT
10. Hukum
    a. Pengenalan sidik jari
    b. Pengenalan foto narapidana

2.2.7 Metoda Normalize Sum-Square Differences(NSSD)

Metoda ini digunakan sebagai sensor untuk mengenali objek, yang mengambil selisih jumlah pixel frame dan background dan dikuadratkan. Nilai NSSD yang didapat akan menunjukan lalu lintas kendaraan

Secara blok diagram, sistem yang dibuat adalah seperti gambar II.26 Sistem ini dibagi menjadi empat bagian, yaitu pemrosesan background image dan frame, perhitungan NSSD, Filtering, serta Thresholding.

Untuk melakukan identifikasi dengan menggunakan metode NSSD, diperlukan background image untuk dibandingkan dengan frame yang akan dideteksi. Background image yang digunakan merupakan image jalan atau lajur dalam keadaan kosong atau tidak ada kendaraan pada lajur yang akan dideteksi. ROI(Region of Interest) diset pada lajur yang akan dideteksi setelah dilakukan proses grayscale sebelumnya. Kemudian, semua nilai pixel yang terdapat dalam detection window dijumlahkan

Hal yang sama juga dilakukan pada setiap frame video. Maksud dari frame video adalah sekumpulan image yang ditampilkan secara berurutan sehingga membentuk suatu gambar bergerak. Sehingga proses grayscale, setting ROI(Region of Interest), dan menjumlahkan nilai pixel dilakukan pada masing-masing frame. Posisi ROI(Region of Interest) pada frame dan pada background image harus sama karena pada posisi inilah yang akan dilakukan pembandingan.

Dengan adanya ROI(Region of Interest) memudahkan proses perhitungan jumlah nilai pixel yang dilakukan oleh komputer. Bilamana tidak menggunakan ROI(Region of Interest), komputer akan melakukan perhitungan pada frame secara keseluruhan. Sedangkan dengan ROI(Region of Interest), komputer cukup menghitung pada ROI(Region of Interest) saja.

2.2.7.1 Perhitungan NSSD

Setelah melakukan proses setting ROI dan mengubah citra menjadi citra gray, langkah selanjutnya menghitung nilai *Normalized Sum Squared Differences* (NSSD) yang terdapat pada setiap frame yang telah dikonversi menjadi *grayscale*. Perhitungan dilakukan di dalam ROI (gambar II.30). Dalam perhitungan NSSD dibutuhkan nilai intensitas piksel (derajat keabuan) citra

background dan citra frame. Langkah-langkah Metode NSSD adalah sebagai berikut:

1. Citra *background* dan nilai pikselnya

   Tentukan citra *background* dan nilai pikselnya. Citra ini harus sudah dalam bentuk citra abu-abu. Contoh diperlihatkan pada gambar II.32
2. Tentukan citra frame dan nilai pikselnya

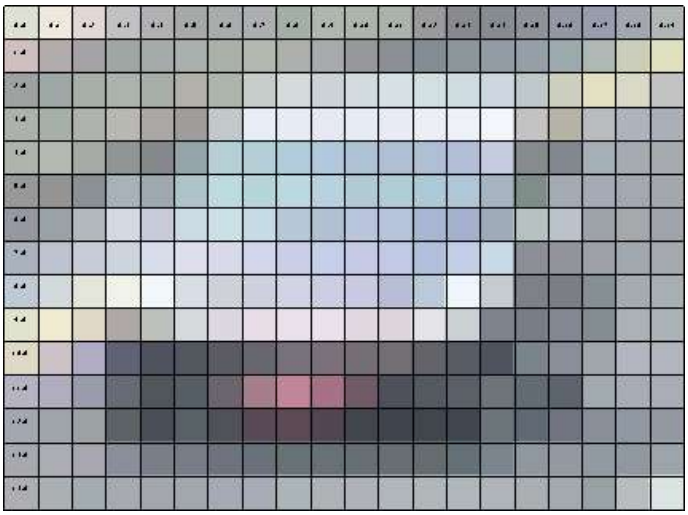
   Tentukan citra *frame* dan nilai pikselnya. Citra ini juga harus sudah dalam bentuk citra abu-abu. Contoh diperlihatkan pada gambar II.33
3. Untuk setiap frame, lakukan pemrosesan setiap piksel yang menggunakan persamaan NSSD berikut:

$$NSSD = \frac{\Sigma \left(P_f(i) - P_b(i) \right)^2}{N}$$

   dengan P_f adalah nilai piksel *frame*, P_b adalah nilai piksel *background*, dan N adalah jumlah piksel yang diproses. Contoh diperlihatkan pada gambar III.33.
4. Tentukan nilai rata-rata NSSD.

2.2.7.2 Contoh Penerapan Pemrosesan NSSD

Contoh penerapan pemprosesan NSSD Secara detail dapat dilihat pada contoh gambar II.26 dengan ukuran lebar 20 piksel dan panjang 15 piksel.

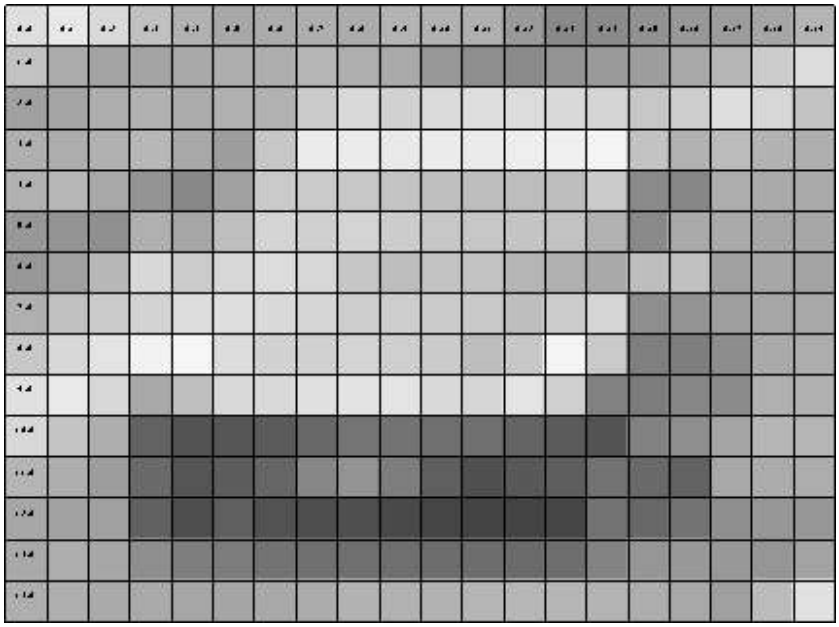


Gambar II.26 Citra Berwarna RGB

Nilai RGB untuk setiap piksel gambar II.26 diperlihatkan pada Gambar II.27. Format nilai piksel adalah (*Red, Green, Blue*).

X / Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	(225, 218, 210)	(236, 231, 225)	(217, 216, 214)	(191, 193, 192)	(171, 175, 174)	(163, 169, 167)	(166, 172, 168)	(168, 177, 172)	(162, 173, 167)	(170, 181, 175)	(168, 178, 170)	(167, 176, 173)	(147, 157, 159)	(131, 142, 148)	(129, 143, 146)	(135, 150, 153)	(141, 154, 160)	(149, 160, 162)	(159, 169, 161)	(174, 182, 169)
1	(196, 193, 186)	(173, 172, 167)	(162, 163, 158)	(161, 163, 160)	(164, 169, 165)	(167, 173, 169)	(167, 173, 171)	(172, 181, 178)	(166, 177, 173)	(161, 172, 168)	(144, 155, 151)	(135, 145, 144)	(133, 142, 147)	(139, 150, 156)	(142, 156, 159)	(147, 161, 162)	(157, 169, 169)	(171, 182, 178)	(194, 204, 196)	(212, 220, 209)
2	(156, 163, 156)	(159, 166, 159)	(166, 173, 165)	(169, 176, 168)	(166, 171, 164)	(170, 176, 172)	(168, 177, 174)	(193, 203, 202)	(206, 216, 217)	(199, 209, 210)	(207, 217, 219)	(211, 220, 225)	(208, 217, 224)	(206, 215, 222)	(200, 211, 217)	(187, 197, 198)	(199, 204, 198)	(218, 221, 212)	(208, 213, 207)	(186, 192, 188)
3	(166, 177, 171)	(163, 174, 166)	(169, 179, 170)	(175, 185, 176)	(160, 170, 162)	(150, 161, 157)	(184, 200, 199)	(220, 238, 240)	(220, 238, 240)	(219, 237, 241)	(221, 238, 245)	(220, 237, 245)	(226, 240, 249)	(229, 244, 251)	(234, 247, 255)	(185, 195, 197)	(172, 177, 171)	(182, 187, 180)	(172, 178, 176)	(166, 175, 174)
4	(168, 178, 170)	(173, 182, 177)	(158, 169, 165)	(139, 151, 149)	(127, 141, 141)	(143, 163, 164)	(175, 205, 205)	(170, 204, 205)	(170, 201, 206)	(168, 198, 206)	(163, 193, 201)	(162, 192, 202)	(163, 191, 202)	(169, 191, 202)	(187, 204, 212)	(129, 142, 148)	(128, 139, 141)	(162, 172, 173)	(162, 172, 174)	(162, 169, 175)
5	(145, 156, 152)	(140, 150, 149)	(136, 146, 147)	(165, 176, 180)	(154, 169, 174)	(171, 192, 197)	(182, 213, 216)	(174, 209, 213)	(179, 213, 222)	(173, 207, 217)	(167, 201, 213)	(168, 200, 213)	(168, 197, 213)	(169, 195, 208)	(160, 178, 188)	(127, 142, 149)	(158, 171, 177)	(160, 171, 177)	(157, 168, 174)	(159, 168, 175)
6	(144, 155, 159)	(151, 162, 166)	(174, 185, 191)	(204, 215, 221)	(190, 203, 211)	(197, 211, 220)	(198, 218, 229)	(190, 214, 226)	(173, 199, 214)	(165, 191, 208)	(171, 196, 216)	(172, 194, 217)	(160, 182, 206)	(155, 176, 195)	(152, 170, 180)	(175, 190, 195)	(181, 192, 196)	(152, 163, 167)	(158, 169, 173)	(154, 165, 169)
7	(166, 177, 179)	(182, 193, 195)	(193, 203, 205)	(201, 208, 214)	(210, 219, 226)	(212, 222, 231)	(205, 217, 229)	(199, 212, 228)	(190, 206, 222)	(189, 206, 224)	(188, 203, 224)	(184, 199, 222)	(172, 188, 211)	(187, 204, 222)	(196, 213, 223)	(133, 146, 152)	(139, 150, 154)	(151, 161, 163)	(158, 169, 173)	(157, 171, 174)
8	(190, 200, 192)	(208, 215, 208)	(224, 227, 220)	(238, 240, 235)	(242, 248, 248)	(210, 218, 221)	(202, 206, 217)	(202, 206, 218)	(201, 207, 221)	(198, 204, 218)	(195, 201, 217)	(178, 185, 203)	(185, 199, 212)	(235, 249, 255)	(191, 202, 208)	(124, 133, 138)	(123, 133, 135)	(133, 144, 146)	(158, 172, 175)	(161, 175, 178)
9	(217, 223, 213)	(229, 232, 225)	(214, 215, 209)	(168, 170, 167)	(184, 190, 188)	(207, 215, 217)	(213, 213, 221)	(223, 220, 231)	(228, 222, 234)	(229, 223, 237)	(215, 213, 227)	(208, 210, 223)	(217, 226, 235)	(195, 206, 212)	(126, 135, 140)	(121, 130, 135)	(130, 139, 144)	(131, 142, 146)	(163, 177, 180)	(164, 178, 181)
10	(213, 213, 213)	(194, 195, 197)	(170, 173, 180)	(104, 108, 117)	(87, 96, 101)	(91, 98, 104)	(102, 99, 108)	(118, 106, 118)	(135, 114, 129)	(134, 116, 130)	(124, 113, 127)	(118, 116, 127)	(103, 107, 118)	(94, 103, 110)	(87, 96, 101)	(124, 135, 139)	(134, 145, 149)	(155, 166, 172)	(171, 182, 188)	(171, 182, 188)
11	(180, 179, 184)	(172, 173, 178)	(154, 158, 169)	(106, 114, 125)	(87, 97, 106)	(96, 105, 110)	(111, 110, 116)	(147, 137, 146)	(168, 148, 160)	(147, 129, 141)	(109, 101, 112)	(91, 91, 99)	(92, 99, 107)	(96, 105, 112)	(111, 122, 126)	(102, 116, 119)	(97, 110, 116)	(156, 169, 175)	(162, 173, 179)	(163, 174, 180)
12	(161, 166, 170)	(157, 162, 168)	(152, 159, 167)	(98, 107, 114)	(81, 92, 98)	(95, 106, 112)	(89, 94, 100)	(92, 91, 97)	(96, 91, 98)	(89, 86, 93)	(79, 82, 89)	(74, 83, 88)	(74, 85, 89)	(72, 86, 89)	(109, 124, 127)	(99, 114, 117)	(111, 124, 130)	(133, 146, 152)	(140, 153, 159)	(140, 153, 159)
13	(162, 170, 173)	(165, 172, 178)	(160, 167, 173)	(136, 145, 150)	(122, 133, 139)	(119, 130, 134)	(113, 122, 127)	(111, 119, 122)	(112, 117, 121)	(112, 117, 121)	(107, 116, 121)	(107, 118, 122)	(103, 117, 118)	(106, 120, 121)	(121, 136, 139)	(138, 153, 156)	(142, 156, 159)	(142, 156, 159)	(140, 154, 157)	(151, 165, 168)
14	(167, 175, 178)	(166, 174, 177)	(162, 171, 176)	(162, 171, 176)	(159, 168, 173)	(158, 167, 172)	(161, 172, 176)	(161, 172, 176)	(168, 178, 180)	(168, 178, 180)	(169, 179, 181)	(172, 182, 184)	(170, 181, 183)	(170, 181, 183)	(167, 181, 184)	(161, 175, 178)	(158, 172, 175)	(148, 162, 163)	(177, 189, 189)	(212, 224, 224)

Gambar II.27 Nilai Piksel Citra RGB



Gambar II.28 Citra Grayscale

Citra RGB pada Gambar II.26 kemudian diubah menjadi citra abu-abu seperti pada gambar II.28 dengan menggunakan persamaan berikut:

$$Grayscale = 0.2125 \times R + 0.7154 \times G + 0.0721 \times B$$

Sehingga untuk masing-masing pikselnya diperoleh melalui proses sebagai berikut:

1. Grayscale(0,0) =0.2125 × 98+0.7154 × 108+0.0721 × 99=105
2. Grayscale(0,1) =0.2125 × 179+0.7154 × 185+0.0721 × 185=183
3. Grayscale(0,2) =0.2125 × 175+0.7154 × 180+0.0721 × 184=179
4. Grayscale(0,3) =0.2125 × 175+0.7154 × 179+0.0721 × 180=178
5. Grayscale(0,4) =0.2125 × 173+0.7154 × 181+0.0721 × 183=179
6. Grayscale(0,5) =0.2125 × 190+0.7154 × 191+0.0721 × 177=189
7. Grayscale(0,6) =0.2125 × 169+0.7154 × 175+0.0721 × 161=172
8. Grayscale(0,7) =0.2125 × 141+0.7154 × 150+0.0721 × 149=148
9. Grayscale(0,8) =0.2125 × 190+0.7154 × 193+0.0721 × 176=191

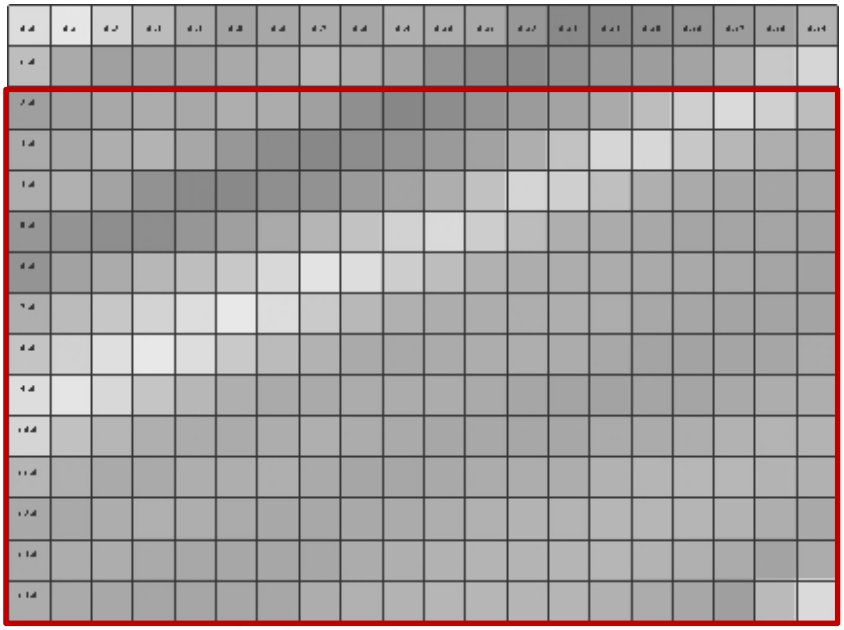
…

300. Grayscale(19,14) =0.2125 × 135+0.7154 × 190+0.0721 × 70=169

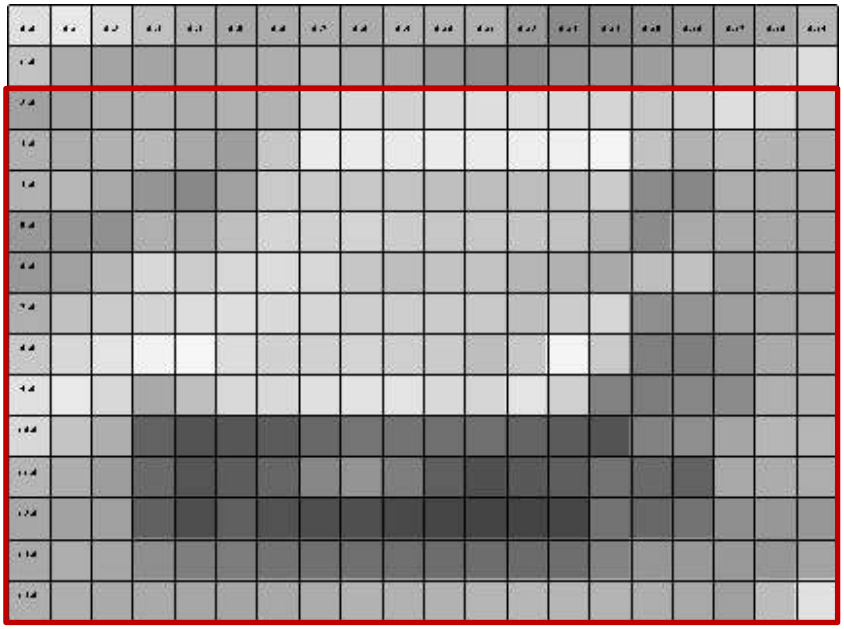
sehingga untuk keseluruhan pikselnya dapat dilihat pada Gambar II.29.

X/Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	218	231	216	192	174	167	170	174	170	178	175	173	155	140	140	147	151	157	166	179
1	193	171	162	162	167	171	171	178	174	169	152	142	140	148	153	158	166	179	201	217
2	161	164	170	173	169	174	174	200	213	206	215	218	215	213	209	194	202	219	211	190
3	174	171	176	182	167	158	196	234	234	233	234	233	237	241	244	193	175	185	176	173
4	175	179	166	148	138	158	198	196	194	192	187	186	185	187	200	139	136	169	170	167
5	153	147	143	173	166	187	206	201	206	200	194	194	191	190	174	139	168	169	166	166
6	152	159	183	213	200	208	214	209	194	186	192	190	179	172	166	187	189	160	166	162
7	174	190	201	206	217	220	215	210	203	203	201	197	186	201	210	143	147	159	166	168
8	197	213	225	239	246	216	205	206	206	203	200	184	196	246	200	131	131	141	169	172
9	221	230	214	169	188	213	213	221	224	225	214	210	224	204	133	128	137	139	174	175
10	213	194	172	107	94	96	100	109	119	120	116	117	106	101	94	132	142	164	180	180
11	179	173	157	113	95	103	110	139	153	133	103	91	98	103	119	113	107	166	171	172
12	165	161	158	105	90	104	93	91	92	87	81	81	82	83	121	111	121	143	150	150
13	168	170	165	143	131	127	120	117	116	116	114	115	114	117	133	150	153	153	151	162
14	173	172	169	169	166	165	169	169	176	176	177	180	178	178	178	172	169	159	186	221

Gambar II.29 Nilai Piksel Citra *gray scale*



Gambar II.30 Citra background grayscale dengan ROI 13x20 piksel



Gambar II.31 Citra grayscale dengan ROI 13x20 piksel

Contoh penerapan pemrosesan NSSD adalah sebagai berikut. Nilai piksel adalah 0 sampai dengan 255 merujuk pada skala derajat keabuan (*grayscale*). Gambar berikut memperlihatkan ROI, citra backround, dan frame yang akan diproses menggunakan algoritma NSSD di atas. Pada Gambar II.30 diperlihatkan ROI dengan luas area sebesar 13x20 piksel. Gambar II.32 dan Gambar II.33 adalah citra *background* dan citra frame yang sesuai dengan ukuran yang sama dengan ROI.

X/Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	159	162	167	172	168	174	172	161	143	135	141	148	156	164	171	188	205	217	209	190
1	169	167	173	179	167	151	140	137	141	147	154	161	174	195	214	215	201	183	173	170
2	173	175	163	147	137	137	143	147	155	163	175	194	212	207	190	176	171	165	167	166
3	150	147	141	142	151	161	168	181	194	209	218	205	188	175	172	168	165	165	166	164
4	148	164	171	182	190	201	217	226	221	205	190	178	174	173	172	170	169	168	165	163
5	174	187	199	210	220	232	221	202	185	176	173	173	173	174	173	167	165	164	164	164
6	197	210	223	233	221	201	187	178	170	169	172	174	175	172	168	164	164	165	166	171
7	221	229	217	197	184	175	170	171	172	173	172	170	166	164	164	165	165	169	172	174
8	214	194	182	175	171	171	173	175	172	170	167	167	167	168	170	171	174	178	179	179
9	182	176	170	170	175	175	174	171	166	168	168	171	173	173	178	181	183	182	181	177
10	170	168	173	175	174	170	170	167	170	172	173	176	180	179	183	182	181	179	174	169
11	171	174	173	170	168	168	168	168	172	176	179	180	182	182	183	181	176	172	163	172
12	172	170	168	164	165	167	167	173	176	179	182	182	182	181	177	171	168	160	185	218

Gambar II.32 Nilai Piksel Citra *Backgound* dengan ROI 13x20 piksel

Berikut adalah nilai piksel citra frame.

X/Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	161	164	170	173	169	174	174	200	213	206	215	218	215	213	209	194	202	219	211	190
1	174	171	176	182	167	158	196	234	234	233	234	233	237	241	244	193	175	185	176	173
2	175	179	166	148	138	158	198	196	194	192	187	186	185	187	200	139	136	169	170	167
3	153	147	143	173	166	187	206	201	206	200	194	194	191	190	174	139	168	169	166	166
4	152	159	183	213	200	208	214	209	194	186	192	190	179	172	166	187	189	160	166	162
5	174	190	201	206	217	220	215	210	203	203	201	197	186	201	210	143	147	159	166	168
6	197	213	225	239	246	216	205	206	206	203	200	184	196	246	200	131	131	141	169	172
7	221	230	214	169	188	213	213	221	224	225	214	210	224	204	133	128	137	139	174	175
8	213	194	172	107	94	96	100	109	119	120	116	117	106	101	94	132	142	164	180	180
9	179	173	157	113	95	103	110	139	153	133	103	91	98	103	119	113	107	166	171	172
10	165	161	158	105	90	104	93	91	92	87	81	81	82	83	121	111	121	143	150	150
11	168	170	165	143	131	127	120	117	116	116	114	115	114	117	133	150	153	153	151	162
12	173	172	169	169	166	165	169	169	176	176	177	180	178	178	178	172	169	159	186	221

Gambar II.33 Nilai Piksel Citra Frame

Selanjutnya untuk kuadrat diferensiasi untuk frame diperlihatkan pada Gambar II.34

X/Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	4	4	9	1	1	0	4	1521	4900	5041	5476	4900	3481	2401	1444	36	9	4	4	0
1	25	16	9	9	0	49	3136	9409	8649	7396	6400	5184	3969	2116	900	484	676	4	9	9
2	4	16	9	1	1	441	3025	2401	1521	841	144	64	729	400	100	1369	1225	16	9	1
3	9	0	4	961	225	676	1444	400	144	81	576	121	9	225	4	841	9	16	0	4
4	16	25	144	961	100	49	9	289	729	361	4	144	25	1	36	289	400	64	1	1
5	0	9	4	16	9	144	36	64	324	729	784	576	169	729	1369	576	324	25	4	16
6	0	9	4	36	625	225	324	784	1296	1156	784	100	441	5476	1024	1089	1089	576	9	1
7	0	1	9	784	16	1444	1849	2500	2704	2704	1764	1600	3364	1600	961	1369	784	900	4	1
8	1	0	100	4624	5929	5625	5329	4356	2809	2500	2601	2500	3721	4489	5776	1521	1024	196	1	1
9	9	9	169	3249	6400	5184	4096	1024	169	1225	4225	6400	5625	4900	3481	4624	5776	256	100	25
10	25	49	225	4900	7056	4356	5929	5776	6084	7225	8464	9025	9604	9216	3844	5041	3600	1296	576	361
11	9	16	64	729	1369	1681	2304	2601	3136	3600	4225	4225	4624	4225	2500	961	529	361	144	100
12	1	4	1	25	1	4	4	16	0	9	25	4	16	9	1	1	1	1	1	9

Gambar II.34 Nilai Piksel *(Frame-Backgound)*^{2} pada frame

Dari Gambar II.33 (*background*) dan Gambar II.34 (*frame*-1) dapat dicari nilai NSSD. Nilai NSSD, dihitung melalui perbandingan antara citra frame yang dinyatakan dengan *f*, dan citra *background* yang dinyatakan dengan *b*. Pada citra frame terdapat objek dengan jumlah piksel. Piksel-piksel yang mengandung unsur objek memiliki nilai kuadrat diferensiasi yang sangat tinggi dibanding piksel lainnya. Perhitungan NSSD dilakukan terhadap piksel-piksel ini. Jumlah piksel dinyatakan dengan N. Berikut adalah perhitungannya.

$$\begin{aligned}
NSSD_{F1} &= ((f(0,7)-B(0,7))^2 + (f(0,8)-B(0,8))^2 + (f(0,9)-B(0,9))^2 + \\
&\quad (f(0,10)-B(0,10))^2 + (f(0,11)-B(0,11))^2 + (f(0,12)-B(0,12))^2 + \\
&\quad \ldots\ldots \\
&\quad (f(11,18)-B(11,18))^2 + (f(11,19)-B(11,19))^2) / 155 \\
NSSD_{F1} &= ((168-158)^2 + (188-158)^2 + (202-173)^2 + \\
&\quad (203-193)^2 + (190-179)^2 + (171-185)^2 + \\
&\quad \ldots\ldots \\
&\quad (121-171)^2 + (117-171)^2) / 155 \\
&= 2475
\end{aligned}$$

2.2.8 UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, danpendokumentasian dari sebuah sistempengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software[3].

Obyek dalam‘software analysis & design’ adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana obyek adalah mobil, manusia, alarm dan lain-lainnya. Tapi obyek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem sepertitabel, database, event, system messages.

Obyek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan obyek tersebut dari obyek lainnya.

Alasan mengapa saat ini pendekatandalam pengembangan software dengan object-oriented, pertama adalahscalability dimana obyek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Keduadynamic modeling, adalah dapat dipakai untuk permodelan sistem dinamis dan real time.

Teknik Dasar OOA/D (Object-Oriented Analysis/Design)

Dalam dunia pemodelan, metodologi implementasi obyek walaupun terikat kaidah-kaidah standar, namun teknikpemilihan obyek tidak terlepas pada subyektifitas *software analyst & designer*. Beberapa obyek akan diabaikan dan beberapa obyek menjadi perhatian untuk diimplementasikan di dalam sistem. Hal ini sah-sah saja karena kenyataan bahwa suatu permasalahan sudah tentu memiliki lebih dari satu solusi. Ada 3(tiga) teknik/konsep dasar dalam OOA/D, yaitu pemodulan (*encapsulation*), penurunan (*inheritance*) dan *polymorphism*.

a. Pemodulan (*Encapsulation*)

   Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan rice cooker, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep information hiding.

b. Penurunan (*Inheritance*)

   Obyek-obyek memiliki banyak persamaan, namun ada sedikit perbedan. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian obyek-obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dapat dikatakan sebagai obyek induk (parent). Sedangkan minibus dikatakan sebagai obyek anak (child), hal ini juga berarti semua operasi yang berlaku pada mobil berlaku juga pada minibus.

c. *Polymorphism*

Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (child) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut polymorphism, teknik atau konsep dasar lainnya adalah ruang lingkup / pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mendefinisikan diagram-diagram sebagai berikut:

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.

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.

Sebuah *use case* dapat meng-*include* fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common* .

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case*

menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan(*atribut*/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *objek* beserta hubungan satu sama lain seperti *containment* , pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private* , tidak dapat dipanggil dari luar class yang bersangkutan
2. *Protected* , hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. *Public* , dapat dipanggil oleh siapa saja

3. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimulus yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

4. *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.

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.

Sama seperti state , standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah

event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *life line* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk *objek boundary, controller* dan *persistent entity*.

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen piranti lunak adalah modul berisi *code* , baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time* , maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface* , yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan

jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

Sebuah node adalah *server*, *workstation* , atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.2.9 Bahasa C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. C# adalah Java versi Microsoft, sebuah bahasa multi flatform yang didesain untuk bisa berjalan di berbagai mesin. C# adalah pemrograman berorientasi Object (OOP). C# memiliki kekuatan bahasa C++ dan portabilitas seperti Java. Fitur-fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti garbage collection, reflection, akar kelas (root class), dan juga penyederhanaan terhadap pewarisan jamak (multiple inheritance)[4].

Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat general-purpose (untuk tujuan jamak), berorientasi objek, modern, dan sederhana. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (hosted system) maupun sistem embedded (embedded system), mulai dari program aplikasi yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada program aplikasi yang sangat kecil.

Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran program aplikasi yang dibuat dengan menggunakan bahasa pemrograman C.

2.2.9.1 Perkembangan C# .NET

C# .NET saat ini telah berkembang hingga versi ke 3, yaitu C# 3.0. Sejak awal kemunculannya, C# .NET telah memberikan sesuatu yang baru di keluarga bahasa pemprograman Microsoft. Dengan struktur bahasanya yang simpel,

modern, dan berbasis OOP, C# .NET telah menjaring banyak programmer untuk menggunakannya. Walaupun C# .NET mirip dengan Java, namun dengan didukung oleh IDE yang menarik dan simpel pada Visual Studio serta dukungannya terhadap metode RAD, C# .NET telah menjadi salah satu bahasa yang fenomenal.

Diantara keluarga bahasa pemprograman Microsoft lainnya, C# .NET merupakan satu-satunya yang benar-benar terintegrasi dengan CLI-nya .NET Framework. Karena C# mempunyai fitur-fitur yang sangat erat kaitannya dengan CLI dan implementasi dari C# seluruhnya ditragetkan pada CLI.

Dengan berkembangnya C# .NET, bahasa ini telah disetujui menjadi ISO Standard dan juga oleh ECMA. Dengan distandarisasinya C# .NET hal ini berarti C# .NET semakin luas jangkauan lingkupnya.

2.2.9.2 Fitur-Fitur Utama C# .NET

Kesederhanannya

a. Pointer telah dihilangkan dari C#.

   Hal ini berarti tidak perlu lagi manajemen memori secara manual yang dapat mempersulit programmer dalam membangun suatu aplikasi.

b. Karena berada dalam lingkungan .NET, maka C# memiliki turunan manajemen memori otomatis dan Garbage Collection.

   Untuk mengganti pointer, sistem manajemen memori otomatis seperti ini digunakan, hal ini berdampak pada manjemen memori yang lebih mudah dan mempermudah pekerjaan programmer.

c. Dukungan tipe data primitif yang lebih banyak seperti misalnya Integer, Floats, dan sebagainya.

   Dengan adanya tipe data primitif yang lengkap, programmer dapat memilih tipe data yang ingin digunakan sesuai dengan kondisi yang ada secara optimal tanpa dibatasi oleh jumlah tipe variabel yang disupport.

d. Nilai integer 0 dan 1 tidak diterima lagi sebagai alternatif nilai Boolean. Nilai Boolean adalah murni true atau false.

Dengan adanya peraturan ini berarti C# .NET telah menjalankan konsistensinya dalam penggunaan sintaks pemprograman. Hal ini dapat mengurangi ambiguitas yang sering terjadi ketika membangun suatu aplikasi.

Modern

a. C# didasarkan pada trend yang berkembang saat ini dan sangat ampuh serta mudah digunakan untuk membangun aplikasi yang bersifat interopable, skalabilitas, dan cepat.
b. C# menyertakan dukungan yang dapat membuat setiap komponennya diintegrasikan kedalam web service yang dapat dipanggil dari aplikasi apa saja dan platform apa saja melalui internet.

Object Oriented

a. C# mendukung enkapsulasi, inheritance, polymorphism, dan interfaces secara penuh.
b. (int, float, double) bukanlah merupakan object dalam bahasa Java, tetapi C# memperkenalkan struktur yang memungkinkan tipe-tipe primitif dapat menjadi object

Type Safe

a. Pada C#, kita tidak dapat melakukan hal-hal tertentu seperti mengkonversi nilai variabel bertipe double ke boolean.
b. Indeks array selalu dari Nol dan batasannya selalu diperhatikan.
c. Tipe-tipe yang menghasilkan nilai overflow dapat dievaluasi.

Interoperability

a. C# menyertakan dukungan pada COM dan aplikasi berbasis windows.
b. User sekarang tidak lagi secara eksplisit mengimplementasikan inteface COM yang tidak diketahui, fitur-fitur seperti itu sudah ada pada C#.
c. C# mengizinkan user untuk menggunakan pointer sebagai blok kode yang bersifat tidak aman.

d. Komponen yang dibuat oleh VB.NET atau bahasa pemrograman lain yang mendukung .NET dapat langsung digunakan oleh C#.

Modular

Kode C# ditulis dengan pembagian masing Class-Class (classes) yang terdiri dari beberapa routines yang disebut sebagai member methods. Class-Class dan metode-metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka kita akan dapat membuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (reusable code).

2.2.9.3 Keunggulan Dan Kekurangan

C# diciptakan dengan harapan menjadi bahasa pemrograman yang simpel, modern, dan juga berbasis OOP. C# berusaha mencapai keungulan yang diharapkan dengan menjadikan developer inti dari Delphi menjadi team leader dari C#. C# menjadi bahasa pemrograman yang unggul karena C# mengadopsi konsep positif dan keunggulan-keunggulan dari bahasa yang lain yang merupakan pendahulu dari C# seperti C, C++, JAVA, dan juga VB. Konsep OOP misalnya, C# berbasiskan bahasa C++ dan sangat kental dengan pengaruh konsep OOP milik Java. Sehingga dengan pengadopsian bahasa pemrograman yang memang bagus dalam hal pembuatan aplikasi berbasis OOP maka C# pun juga menjadi bahasa yang sering dipakai oleh programmer. Selain itu sokongan dari Microsoft dan juga tool yang ada, serta tidak lupa IDE yang baik membuat C# dapat bersaing dengan bahasa pemrograman lainya.

Dengan segala keunggulan C# .NET mulai dari manajemen memorinya yang lebih baik, standarisasi oleh ECMA dan ISO, dukungan penuh dari vendor dan lain-lain, C# .NET memiliki kelemahan yang mirip dengan keluarga bahasa pemprograman lainnya yang berada di bawah .NET Framework. C# .NET membutuhkan platform yang akan menjalankan programnya sudah terinstal dengan .NET Framework yang besar filenya semakin lama semakin memberatkan end-user.

C# .NET juga mengalami masalah mengenai penggunaan resource komputernya yang lebih banyak daripada bahasa pemprograman lainnya. Hal ini berdampak pada tingginya requirement yang harus dipenuhi oleh suatu platform agar dapat menajalankan aplikasi C# .NET

2.2.10 Mysql

MySQL ialah sebuah perangkat lunak sistem untuk memanajemen basis data SQL atau sering juga disebut DBMS (Databases Management System). MySQL adalah DBMS yang multithread, multiuser dengan perkiraan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL sebagai perangkat lunak gratis, dibawah lisensi GNU GPL[5].

Tidak sama hal nya dengan Apache, dimana perangkat lunak dikembangkan oleh komunitas umum. MySQL dimiliki dan d siponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang di distribusikan secara gratis dibawah lisensi GPL. SQL adalah sebuah konsep pengoperasian basisdata untuk pemilihan atau seleksi dan pemasukan data, yang pengoperasiannya dapat dengan mudah secara otomatis.

Sebagai basis data, MySQL mendukung operasi basis data transaksional maupun non-transaksional, pada operasi transaksional MySQL lebih unggul dari kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas, yaitu dapat berjalan melintasi multiplatform dan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka atau open source, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user, yaitu dapat digunakan oleh beberapa pengguna atau user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', memiliki kecepatan yang cepat dan stabil dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data, memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi, memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABEL, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.11 *AForge.Net Framework*

AForge.Net merupakan *framework open source* C# yang dirancang untuk pengembangan dan penelitian di bidang *computer vision, artifial intelligence, image processing, neural network, genetic algorithms, fuzzy logic, machine learning, robotics,* dan lain-lain[7].

Framework ini diterbitkan di bawah lisensi LGPLv.3 dan terdiri dari set *library* dan contoh aplikasi, yang menunjukkan karakternya, yaitu sebagai berikut:

1. *AForge.Imaging - library with image processing routines and filters*
2. *AForge.Vision - computer vision library*
3. *AForge.Video - set of libraries for video processing*
4. *AForge.Neuro - neural networks computation library*
5. *AForge.Genetic - evolution programming library*
6. *AForge.Fuzzy - fuzzy computations library*
7. *AForge.Robotics - library providing support of some robotics kits*
8. *AForge.MachineLearning - machine learning library*

Beberapa contoh dari *library imaging processing* dari AForge antara lain:

1. Color Filtering

   Color filtering merupakan filter yang digunakan untuk memfilter warna yang di inginkan pada gambar atau citra berdasarkan nilai komponen pada RGB. Kombinasi RGB inilah yang kemudian dijadikan filter sebagai penentu sebuah warna diloloskan atau tidak.

2. ThresholdedDifference Class

   ThresholdedDifference menghasilkan hasil yang sama seperti menerapkan Difference filter dan kemudian Threshold filter, perbedaan antara dua gambar. Hasil dari rutin pengolahan gambar ini mungkin berguna dalam aplikasi deteksi gerak untuk menemukan perbedaan yang signifikan.

3. Grayscale Filter

Grayscale filter digunakan untuk mengubah citra berwarna menjadi citra gray-scale, hal ini digunakan untuk menyederhanakan model citra. Seperti telah dijelaskan di depan, citra berwarna terdiri dari 3 layer matrik yaitu R-layer, G-layer dan B-layer. Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga layer di atas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Sehingga konsep itu diubah dengan mengubah 3 layer di atas menjadi 1 layer matrik gray-scale dan hasilnya adalah citra gray-scale. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra gray scale dengan nilai s, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r, g dan b sehingga dapat dituliskan menjadi:

$$s = \frac{r + g + b}{3}$$

Untuk mencoba proses konversi citra berwarna menjadi citra gray-scale ini dapat dibuat dengan koefisien konversi adalah: Merah(r): 0,2125; Hijau(g): 0,7154; Biru(b): 0,0721.

4. Blob Counter

Blob counter digunakan untuk menghitung obyek, memfilter obyek, mengekstrak obyek dan mendapatkan dimensi obyek yang ada digambar, yang dipisahkan oleh latar belakang hitam dengan menggunakan prinsip algoritma pelabelan komponen terhubung. Algoritma ini memperlakukan semua pixel dengan nilai kurang atau sama dengan batas ambang background sebagai latar belakang, sedangkan pixel dengan nilai yang lebih maka diperlakukan sebagai pixel obyek.

Contoh penggunaan class BlobCounter

```
C#
// create an instance of blob counter algorithm
BlobCounter bc = new BlobCounter( );
// process binary image
bc.ProcessImage( image );
Rectangle[] rects = bc.GetObjectsRectangles( );
// process blobs
foreach ( Rectangle rect in rects )
{
    // ...
}
```