BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Institut Pemerintahan Dalam Negeri (IPDN) adalah salah satu Lembaga Pendidikan Tinggi Kedinasan dalam lingkungan Departemen Dalam Negeri Republik Indonesia, yang bertujuan mempersiapkan kader pemerintah, baik di tingkat daerah maupun di tingkat pusat. Pada 10 Oktober 2007, dalam sebuah sidang kabinet, Presiden Susilo Bambang Yudhoyono memutuskan untuk menggabungkan STPDN dengan Institut Ilmu Pemerintahan (IIP) menjadi IPDN menyusul terungkapnya kasus kekerasan yang terjadi di STPDN.

2.1.1 Sejarah nstansi

Penyelenggaraan pendidikan kader pemerintahan di lingkungan Departemen Dalam Negeri yang terbentuk melalui proses perjalanan sejarah yang panjang. Perintisiannya dimulai sejak zaman pemerintahan Hindia Belanda pada tahun 1920, dengan terbentuknya sekolah pendidikan Pamong Praja yang bernama Opleiding School Voor Inlandshe Ambtenaren (OSVIA) dan Middlebare Opleiding School Voor Inlandsche Ambtenaren (MOSVIA). Para lulusannya sangat dibutuhkan dan dimanfaatkan untuk memperkuat penyelenggaraan pemerintahan Hindia Belanda. Dimasa kedudukan pemerintah Hindia Belanda, penyelenggaraan pemerintahan Hindia Belanda dibedakan atas pemerintahan yang langsung dipimpin oleh kaum atau golongan pribumi yaitu Binnenlands Bestuur Corps (BBC) dan pemerintahan yang tidak langsung dipimpin oleh kaum atau golongan dari keturunan Inlands Bestuur Corps (IBC).

Pada masa awal kemerdekaan RI, sejalan dengan penataan system pemerintahan yang diamanatkan oleh Undang Undang Dasar 1945, kebutuhan akan tenaga kader pamong praja untuk melaksnakan tugas-tugas pemerintahan baik pada pemerintah pusat maupun daerah semakin meningkat sejalan dengan tuntutan perkembangan penyelenggaraan pemerintahannya. Untuk memenuhi kebutuhan akan kekurangan tenaga kader pamong praja, maka pada tahun 1948

dibentuklah lembaga pendidikan dalam lingkungan Kementrian Dalam Negeri yaitu Sekolah Menengah Tinggi (SMT) Pangreh Praja yang kemudian berganti nama menjadi Sekolah Menengah Pegawai Pemerintahan Administrasi Atas (SMPAA) di Jakarta dan Makassar.

Pada perkembangan selanjutnya, lulusan APDN dinilai masih perlu ditingkatkan dalam rangka upaya lebih menjamin terbentuknya kader-kader pemerintahan yang ”qualified leadership and manager administrative” terutama dalam menyelenggarakan tugas-tugas urusan pemerintahan umum. Kebutuhan ini mendorong pemerintah untuk menyelenggarakan pendidikan aparatur di lingkungan Departemen Dalam Negeri setingkat Sarjana, maka dibentuklah Institut Ilmu Pemerintahan (IIP) yang berkedudukan di Kota Malang Jawa Timur berdasarkan Keputusan Bersama Menteri Dalam Negeri dan Menteri Pendidikan dan Kebudayaan No. 8 Tahun 1967, selanjutnya dikukuhkan dengan Keputusan Presiden Nomor 119 Tahun 1967. Peresmian berdirinya IIP di Malang ditandai dengan peresmian oleh Presiden Soekarno pada tanggal 25 Mei 1967.

Pada tahun 1972 Institut Ilmu Pemerintahan (IIP) yang berkedudukan di Malang Jawa Timur dipindahkan ke Jakarta melalui Keputusan Menteri Dalam Negeri Nomor 94 Tahun 1972. Pada tanggal 9 Maret 1972, kampus IIP yang terletak di Jakarta di resmikan oleh Presiden Soeharto yang dinyatakan : “Dengan peresmian kampus Institut Ilmu Pemerintahan, mudah-mudahan akan merupakan kawah candradimukanya Departemen Dalam Negeri untuk menggembleng kader-kader pemerintahan yang tangguh bagi Negara Kesatuan Republik Indonesia”.

Seiring dengan pembentukan IIP yang merupakan peningkatan dari APDN Nasional di Malang, maka untuk penyelenggaraan pendidikan kader pada tingkat akademi, Kementrian Dalam Negeri secara bertahap sampai dengan dekade tahun 1970-an membentuk APDN di 20 Provinsi selain yang berkedudukan di Malang, juga di Banda Aceh, Medan, Bukittinggi, Pekanbaru, Jambi, Palembang, Lampung, Bandung, Semarang, Pontianak, Palangkaraya, Banjarmasin, Samarinda, Mataram, Kupang, Makassar, Menado, Ambon dan Jayapura.

Pada tahun 1988, dengan pertimbangan untuk menjamin terbentuknya wawasan nasional dan pengendalian kualitas pendidikan Menteri Dalam Negeri

Rudini melalui Keputusan No. 38 Tahun 1988 Tentang Pembentukan Akademi Pemerintahan Dalam Negeri Nasional. APDN Nasional kedua dengan program DIII berkedudukan di Jatinangor, Sumedang Jawa Barat yang peresmiannya dilakukan oleh Mendagri tanggal 18 Agustus 1990. APDN Nasional ditingkatkan statusnya berdasarkan Kepres No. 42 Tahun 1992 tentang Sekolah Tinggi Pemerintahan Dalam Negeri, maka status APDN menjadi STPN dengan program studi D III yang diresmikan oleh Presiden RI pada tanggal 18 Agustus 1992. Sejak tahun 1995, bertititk tolak dari keinginan dan kebutuhan untuk lebih mendorong perkembangan karier sejalan dengan peningkatan eselonering jabatan dalam sistem kepegawaian Republik Indonesia, maka program studi ditingkatkan menjadi program D IV. Keberadaan STPDN dengan pendidikan profesi (program DIV) dan IIP yang menyelenggarakan pendidikan akademik program sarjana (Strata I), menjadikan Departemen Dalam Negeri memiliki dua (2) Pendidikan Pinggi Kedinasan dengan lulusan yang sama dengan golongan III/a.

Kebijakan Nasional mengenai pendidikan tinggi sejak tahun 1999 antara lain yang mengatur bahwa suatu Departemen tidak boleh memiliki dua atau lebih perguruan tinggi dalam menyelenggarakan keilmuan yang sama, maka mendorong Departemen Dalam Negeri untuk mengintegrasikan STPDN ke dalam IIP . Usaha pengintegrasiaan STPDN kedalam IIP secara intensif dan terprogram sejak tahun 2003 sejalan dengan dikeluarkannya UU No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional. Pengintegrasian terwujud dengan ditetapkannya Keputusan Presiden Nomor 87 Tahun 2004 tentang Penggabungan STPDN ke dalam IIP dan sekaligus merubah nama IIP menjadi Institut Ilmu Pemerintahan (IPDN). Tujuan penggabungan STPDN ke dalam IIP tersebu, selain untuk memenuhi kebijakan pendidikan nasional juga untuk meningkatkan efektivitas penyelenggaraan pendidikan kader pamong praja di lingkungan Departemen Dalam Negeri. Kemudian Kepres No. 87 Tahun 2004 ditindak lanjuti dengan Keputusan Mendagri No.892.22-421 tahun 2005 tentang Pelaksanaan Penggabungan dan Operasional Institut Pemerintahan Dalam Negeri, disertai dengan Peraturan Menteri Dalam Negeri No. 29 Tahun 2005 Tentang Organisasi

dan Tata Kerja IPDN dan Peraturan Menteri Dalam Negeri 43 Tahun 2005 Tentang Statuta IPDN serta peraturan pelaksanaan lainnya.

Pada Tahun 2007 ini, IPDN dikembalikan lagi menjadi Institut Ilmu Pemerintahan (IIP) yang memiliki beberapa Program D-IV di beberapa Regional, diantaranya Regional Jawa-Bali, Sumatera, Sulawesi, Kalimantan, dan Papua.

2.1.2 Lambang Instansi

Gambar 2.1 merupakan lambang dari Institut Pemerintahan Dalam Negeri (IPDN).



Gambar 2.1 Lambang IPDN

Makna bentuk dan motif yang terdapat dalam logo ini adalah :

1. Bintang warna kuning melambangkan Pancasila
2. Kapas warna putih melambangkan keadilan
3. Daun kapas warna hijau melambangkan kesejukan dan ketentraman
4. Padi warna kuning melambangkan kemakmuran.
5. Kombinasi bunga kapas dan daunnya berjumlah 17 melambangkan tanggal Proklamasi 17 Agustus 1945 berdirinya Negara Kesatuan Republik Indonesia.
6. Padi berjumlah 45 bermakna tahun kemerdekaan Republik Indonesia 1945.
7. Roda kemudi melambangkan pemerintahan.
8. Delapan jari roda kemudi melambangkan bulan lahirnya Proklamasi, dan melambangkan 8 penjuru angin yang dimaknai sebagai ke wilayahan, pemerintahan daerah dan Bhinneka Tunggal Ika.

9. Lambang yang berbentuk kelopak bunga lotus (teratai), bermakna kearifan.
10. Buku melambangkan sumber pengetahuan.
11. Warna biru laut yang mendasari lambang dimaknai sebagai tanggungjawab, ketangguhan, ketenangan dan inovasi yang tinggi.
12. Angka 2004 melambangkan tahun berdirinya IPDN.
13. Among, melaksanakan fungsi pamong yang berarti mengasuh dan mengemong menurut sistem among : Ing Ngarso Sung Tulodho, Ing Madyo Mangun Karso, Tutwuri Handayani.
14. Praja artinya Peserta Didik (termasuk mahasiswa) IPDN.
15. Dharma artinya melaksanakan kewajiban, peraturan, kebenaran.
16. Kata-kata : Among Praja Dharma Nagari secara keseluruhan berarti IPDN mengemong Praja supaya setia pada kewajiban untuk mengabdi kepada Bangsa dan Negara.

2.1.3 Visi dan Misi

Berikut adalah visi dan misi dari Institut Pemerintahan Dalam Negeri (IPDN):

2.1.3.1 Visi

Dengan memperhitungkan segala tantangan, kendala dan peluang yang akan dihadapi sebagai akibat dari pengaruh lingkungan strategis serta dengan tidak mengabaikan kondisi Pendidikan Nasional pada saat ini beserta permasalahannya, Visi penyelenggaraan IPDN adalah, **"Unggul dalam menyiapkan kader pamong praja yang berwawasan negarawan, ilmuwan, professional dan demokratis dengan berdasarkan ketaqwaan kepada Tuhan Yang Maha Esa dan dengan memperhatikan lingkungan local, nasional, dan global."** Makna dari visi di atas adalah bahwa dengan melalui penyelenggaraan pendidikan, dilakukan pemberdayaan pemerintahan dalam negeri yang berkualitas, guna mendukung penyelenggaraan pemerintahan di daerah dan di pusat serta membrikan pelayanan prima pada masyarakat luas. Dari visi IPDN di

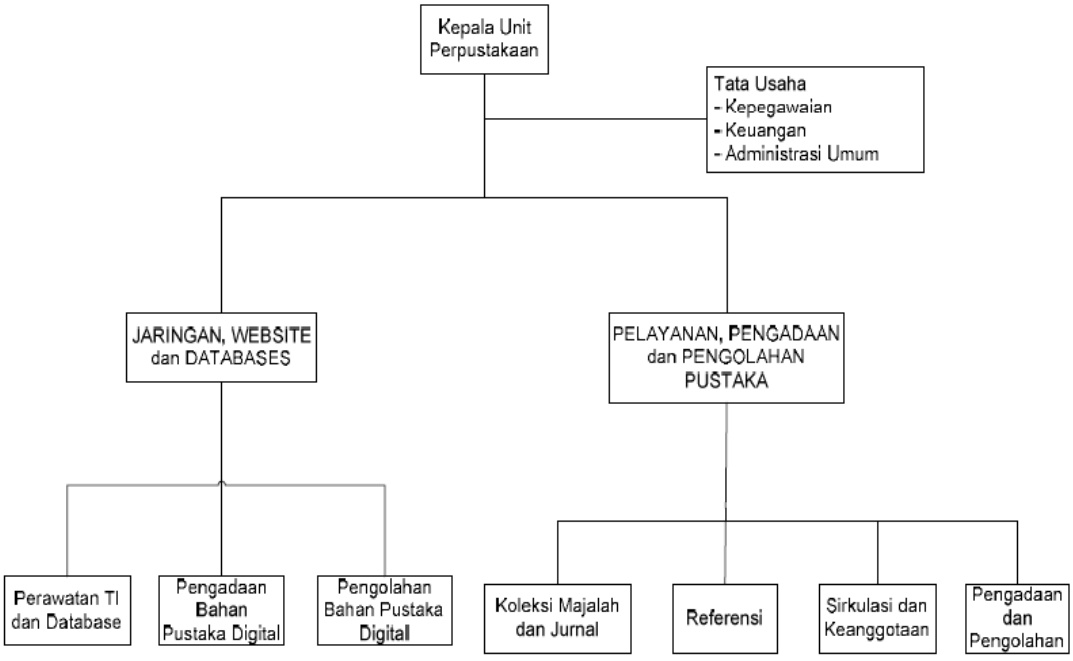
atas, terdapat tiga kompetensi dasar yang harus dimiliki oleh lulusan IPDN yaitu: kepemimpinan (stewardship) Kenegarawanan (statelmanship). Ketiga kompetensi di atas menjadi produk unggulan bagi lulusan atau alumni Institut Pemerintahan Dalam Negeri.

2.1.3.2 Misi

Berdasarkan visi yang telah dipaparkan diatas maka Misi IPDN adalah: "Meningkatkan kualitas peserta didik sesuai dengan tuntutan kebutuhan penyelenggara pemerintahan dan pembangunan yang berwawasan budaya dan lingkungan serta meletakan ladasan pembentukan watak dan kepribadian pengalaman, nilai-nilai agama, budi pekerti yang luhur, memiliki wawasan dan berjiwa kebangsaan serta penguasaan ilmu pengetahuan dan teknologi"

2.1.4 Struktur Organisasi

Gambar 2.2 adalah struktur organisasi di Perpustakaan IPDN Jatinangor:



Gambar 2.2 Struktur Organisasi Perpustakaan IPDN

Tugas dan Wewenang

Tugas dan Wewenang dari struktur organisasi Perpustakaan Institut Pemerintahan Dalam Negeri (IPDN) Jatinangor adalah sebagai berikut:

1. Kepala Perpustakaan

   Tugas dan tanggung jawab kepala perpustakaan adalah:

   a. Membuat perencanaan Strategi kegiatan-kegiatan perpustakaan
   b. Mengkoordinasi semua kegiatan pelayanan perpustakaan yang ada di lingkungan perguruan tinggi
   c. Menjalin kerjasama dengan instansi terkait baik di dalam maupun di luar negeri dalam rangka menyelenggarakan pelayanan perpustakaan
   d. Mengelola sumber-sumber informasi penunjang kegiatan akademik yang ada di lingkungan Institut
   e. Melakukan pembinaan dan usaha pengembangan sumber daya manusia yang terdiri dari pustakawan dan pegawai perpustakaan
   f. Membuat evaluasi pelaksanaan kegiatan perpustakaan
   g. Mengkoordinasi dan membina pustakawan di lingkungan perpustakaan
   h. Mengkoordinasi dan mengevaluasi kegiatan-kegiatan kepala bidang pelayanan
   i. Mengkoordinasi dan mengevaluasi kegiatan-kegiatan kepala bidang jaringan dan database
   j. Menyusun kegiatan rencana dan anggaran tahunan perpustakaan
   k. Merencanakan, memonitor dan mengevaluasi kegiatan pelayanan perpustakaan se-Institut
   l. Atas perintah pimpinan menjalankan tugas-tugas/kegiatan lain yang dapat mendukung keberhasilan perpustakaan
   m. Membuat laporan secara periodik kepada pimpinan IPDN

2. Tata Usaha

   Tugas dan tanggung jawab bagian tata usaha adalah:

   a. Melakukan koordinasi kegiatan tata usaha perpustakaan

b. Membuat sistem penulisan surat, laporan, pembukuan dan inventarisasi kantor, serta sistem pengarsipan
c. Melakukan koordinasi pengelolaan teknis kegiatan perpustakaan
d. Mengkoordinasikan aset-aset perpustakaan
e. Melakukan koordinasi pengelolaan teknis kegiatan perpustakaan
f. Mengkoordinasikan aset-aset perpustakaan

2. Bidang Jaringan, Website dan Database

Tugas dan tanggung jawab di bidang Jaringan, Website dan Database adalah:

a. Mengkoordinasi sumber daya manusia bidang teknologi informasi perpustakaan untuk mengelola sistem dan objek-objek informasi dalam format digital
b. Melakukan standarisasi format database yang digunakan di tiap-tiap perpustakaan di lingkungan Institut
c. Merancang infrastruktur jaringan internet dan intranet di semua perpustakaan di lingkungan IPDN
d. Meningkatkan dan mengelola informasi dalam jaringan dan database;
e. Merencanakan software sistem informasi perpustakaan terpadu untuk semua perpustakaan di lingkungan IPDN
f. Membuat laporan berkala ke Pimpinan Perpustakaan

3. Bidang Pengadaan, Pelayanan dan Pengolahan Perpustakaan

Tugas dan tanggung jawab bidang Pengadaan, Pelayanan dan Pengolahan Perpustakaan adalah:

a. Mengkoordinasi jalannya pengelolaan dan peningkatan bahan pustaka
b. Menginventarisasi dan mencatat bahan pustaka
c. Mengklasifikasikan dan mengkatalogkan bahan pustaka
d. Memelihara dan merawat aset-aset perpustakaan
e. Menerima kunjungan perpustakaan
f. Mengkoordinasi jalannya pelayanan di perpustakaan pusat, perpustakaan fakultas, perpustakaan lembaga, dan perpustakaan pusat studi di lingkungan IPDN.

g. Membina, membimbing serta mengarahkan sumber daya manusia perpustakaan di bidang pelayanan perpustakaan
h. Merancang inovasi pelayanan untuk keperluan kepuasan pemakai serta meningkatkan citra Perpustakaan IPDN
i. Membuat usulan pengadaan prasarana untuk keperluan pelayanan
j. perpustakaan;
k. Membuat laporan berkala kepada pemimpin perpustakaan.

2.2 Landasan Teori

Landasan teori dari penulisan skripsi ini menguraikan proses analisis sistem serta mendukung proses integrasi teknologi RFID pada sistem informasi perpusatakaan.

2.2.1 Perpustakaan

Perpustakaan berasal dari kata *pustaka* artinya buku atau kitab. Dalam bahasa inggris perpustakaan perpustakaan disebut *library*, dalam bahasa Belanda disebut *bibliotheek*, dalam bahasa Perancis *bibliotheque*, dalam bahasa Spanyol dan Portugis *bibliotheca*. Dalam sebutan dari berbagai bahasa tersebut dapat disimpulkan bahwa sebutan *library* terdapat akar yang sama. Akar *library* adalah *liber* (bahasa Latin) artinya buku, sedangkan akar kata *bibliotheek* adalah *biblos* juga artinya sama buku (Yunani). Dengan demikian istilah perpustakaan selalu dikaitkan dengan buku. Sebagai contoh definisi perpustakaan yang dikaitkan dengan buku mengatakan bahwa perpustakaan adalah sebuah ruangan, bagian atau subbagian dari sebuah gedung ataupun gedung itu sendiri yang digunakan untuk menyimpan buku, biasanya disimpan menurut tata susunan tertentu serta digunakan untuk anggota perpustakaan. Definisi lain mengacu pada kumpulan buku atau akomodasi fisik tempat buku dikumpulsusunkan untuk keperluan bacaan, studi, kenyamanan maupun kesenangan. Jadi dalam ancangan tempat ini, konsep perpustakaan mengacu pada bentuk fisik tempat penyimpanan buku (dalam arti luas) maupun sebagai kumpulan buku yang disusun untuk keperluan pembaca.[2]

Dalam arti tradisional, perpustakaan adalah sebuah koleksi buku dan majalah. Walaupun dapat diartikan sebagai koleksi pribadi perseorangan, namun perpustakaan lebih umum dikenal sebagai sebuah koleksi besar yang dibiayai dan dioperasikan oleh sebuah kota atau institusi, dan dimanfaatkan oleh masyarakat yang rata-rata tidak mampu membeli sekian banyak buku atas biaya sendiri.

Dalam Undang-undang No. 43 Tahun 2007 pasal 1 menyebutkan bahwa perpustakaan adalah institusi yang mengumpulkan pengetahuan tercetak dan terekam, pengelolaannya dengan cara khusus guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka.

Sedangkan perpustakaan perguruan tinggi adalah perpustakaan yang berada dibawah pengawasan dan dikelola oleh perguruan tinggi dengan tujuan utama membantu perguruan tinggi mencapai tujuannya. Dalam pengertian ini, perguruan tinggi adalah universitas, fakultas, jurusan, institute, sekolah tinggi dan akademi serta berbagai badan bawahannya seperti lembaga penelitian[2]. Perpustakaan perguruan tinggi ini merupakan unit pelaksana teknis (UPT) yang bersama-sama dengan unit lain melaksanakan Tri Dharma PT (Perguruan Tinggi) melalui menghimpun, memilih, mengolah, merawat serta melayani sumber informasi kepada lembaga induk khususnya dan masyarakat akademis pada umumnya. Perpustakaan di Perguruan Tinggi sering diibaratkan sebagai jantungnya perguruan tinggi (*heart of university*), maka keberadaannya harus ada agar dapat memberikan layanan kepada *civitas* akademik sesuai dengan kebutuhan akademik.

Pada saat ini, dengan semakin banyak koleksi dan penemuan media baru selain buku untuk menyimpan informasi banyak perpustakaan kini juga merupakan tempat penyimpanan dan akses ke berkas, cetak atau hasil seni lainnya, microfilm, microfiche, tape audio, CD, LP, tape video, DVD, dan menyediakan fasilitas umum untuk mengakses gudang data CD-ROM dan internet. Oleh karena itu perpustakaan modern telah didefinisikan kembali sebagai tempat untuk mengakses informasi dalam format apa pun, apakah informasi itu disimpan dalam gedung perpustakaan tersebut atau tidak. Dalam perpustakaan modern ini selain kumpulan buku tercetak, sebagian buku dan koleksinya ada

dalam perpustakaan digital (dalam bentuk data yang bisa diakses lewat jaringan komputer).

2.2.1.1 Periodisasi Perpustakaan

Perpustakaan merupakan perantara masyarakat. oleh karena itu, perkembangan perpustakaan tidak dapat dilepaskan dari perkembangan masyarakat. Perkembangan masyarakat tercermin dalam sejarah masyarakat, kadang-kadang dalam sejarah negara. Dengan demikian,sejarah perpustakaan di Indonesia tidak terlepas dari sejarah Indonesia. Sejarah Indonesia dapat dibagi menjadi beberapa periode berikut :

1. Zaman kerajaan lokal
2. Zaman kerajaan Islam
3. Zaman Hindia Belanda
4. Zaman Jepang
5. Periode pasca 1945, acapkali dibagi lagi menjadi :
    a. Periode 1945-1959
    b. Periode 1959-1965
    c. Periode 1965- sekarang

Pada pembagian di atas, tahun 1950 merupakan awal ancangan karena pada waktu itu pemerintah RI mulai menyebarkan perpustakaan, khususnya perpustakaan umum dengan nama taman perpustakaan rakjat ke seluruh indonesia. Perkembangan perpustakaan umum yang mula.mula menggembirakan itu akhirnya berakhir tragis dengan runtuhnya berbagai taman pustaka rakjat yang didirikan pada tahun 1950-an. Tonggak kebangkitan dimulai pada tahun 1969, dengan pembangunan lima tahun (pelita) pertama. Saat itu, kegiatan perpustakaan tercakup di dalam rencana pembangunan hingga sekarang.[2]

2.2.1.2 Peran dan Fungsi Perpustakaan

Perpustakaan berperan sebagai upaya untuk memelihara dan meningkattkan efisiensi dan efektifitas proses belajar-mengajar. Perpustakaan yang terorganisir secara baik dan sisitematis, secara langsung atau pun tidak langsung dapat memberikan kemudahan bagi proses belajar mengajar di sekolah tempat perpustakaan tersebut berada. Hal ini, trekait dengan kemajuan bidang pendidikan dan dengan adanya perbaikan metode belajar-mengajar yang dirasakan tidak bisa dipisahkan dari masalah penyediaan fasilitas dan sarana pendidikan.

Sedangkan fungsi dari perpustakaan adalah :

1. Penyimpanan, artinya perpustakaan bertugas menyimpan pustaka yang diterimanya.
2. Penelitian, artinya perpustakaan menyediakan buku (dalam arti luas) bagi keperluan penelitian, penelitian ini mencakup penelitian sederhana hingga penelitian rumit, mulai dari penelitian terapan hingga penelitian murni. Untuk melaksanakan fungsi tersebut, perpustakaan bertugas antara lain menyediakan jasa bibliografis, artinya menyusun daftar buku mengenai sebuah subjek, kadangkala disertai dengan sari keterangan artikel tersebut; jasa peminjaman artinya perpustakaan meminjamkan buku untuk anggotanya; dan jasa penelusuran, yaitu jasa pencarian informasi bagi pemakai.
3. Informasi, artinya perpustakaan memberikan informasi mengenai suatu masalah kepada pemakai. Seringkali informasi ini diberikan tanpa diminta bila perpustakaan menganggap informasi tersebut sesuai dengan minat pemakai.
4. Pendidikan, artinya perpustakaan merupakan tempat belajar seumur hidup, terutama bagi mereka yang telah meninggalkan bangku sekolah. Kalau siswa, pelajar dan mahasiswa masih dapat menggunakan perpustakaan masing-masing dan pekerja atau karyawan dapat memanfaatkan jasa perpustakaan khusus, maka perpustakaan umum menyediakan bagi seluruh lapisan masyarakat. Bagi wiraswasta, petani, meraka yang putus sekolah,

pensiunan dan sejenisnya, satu-satunya perpustakaan yang terbuka bagi mereka hanyalah perpustakaan umum.

5. Kultural, artinya petugas meningkatkan nilai budaya dan apresiasi budaya pada kalangan masyarakat melalui penyediaan pustaka. Pustaka ini menjadi bahan ajar dan bacaan. Bacaan ini ada yang bersifat serius, namun tidak sedikit pula yang bersifat hiburan. Dengan membaca, maka masyarakat diperkaya dengan berbagai ilmu pengetahuan. Membaca, terutama membaca untuk tujuan rekreasi, merupakan rekreasi kultural yang mampu menambah khasanah rihaniah pembaca. Perpustakaan juga bertugas melakukan pameran sesuai dengan peristiwa dan lingkungan budaya tempat perpustakaan berada.

2.2.1.3 Model Pelayanan Perpustakaan

Ada empat model pelayanan perpustakaan:

1. Koleksi perpustakaan ada pada kampus cabang.
2. Berpusat pada layanan pinjam antar perpustakaan, resource sharing, dan mahasiswa dapat menggunakan perpustakaan afiliasi.
3. Pengiriman materi dari instuasi induk kepada para mahasiswa
4. Berhubungan dengan penggunaan teknologi untuk mengakses sumber-sumber informasi elektronik

2.2.1.4 Sejarah Perkembangan Perpustakaan

Perkembangan perpustakaan tidak dapat dipisahkan di sejarah manusia karena perpustakaan merupakan produk manusia. Dalam sejarahnya, manusia mula-mula tidak menetap sebagai mengembara dari satu tempat ke tempat yang lain. Kehidupan seperti ini sering disebut kehidupan nomaden. Manusia mencari makan dari alam sekitarnya, sedangkan untuk keperluan ternaknya ia mencari sumber air serta rumput. Manusia mulai berusaha menggarap lahan yang ada disekitarnya, untuk keperluan daging manusia memburu binatang yang ada disekitarnya. Kehidupan berburu ini tidak beranjak jauh dari kehidupan nomaden. Dalam pengembarannya serta dari kehidupan bertaninya, manusia memperoleh

pengalaman bahwa bila dia member tanda pada sebuah batu, pohon, papan, lempengan serta benda lainnya, ternyata manusia dapat menyampaikan berita ke manusia lainnya. Pesan ini dipahatkan pada batu atau pohon atau benda lainnya. Selama itu manusia berhubungan dengan manusia lain melalui bahasa lisan maupun bahasa isyarat. Setelah menggunakan berbagai tanda yang dipahatkan pada pohon ataupun batu ataupun benda lainnya, manusia mulai berkomunikasi dengan kelompok lain melalui bahasa tulisan.

Adanya tulisan tersebut dapat membantu daya ingat manusia daya ingat manusia kini manusia dapat melihat "catatannya" pada pohon, batu, dan lempengan. Pesan dalam berbagai pahatan itu dapat diteruskan ke generasi berikutnya. Bila kegiatan memberi tanda pada berbagai benda itu dilakukan dari satu generasi ke generasi yang berikutnya maupun dari suku satu ke suku lainnya maka banyak dugaan bahwa perpustakaan dalam bentuknya yang sangat sederhana sudah mulai dikenal ketika manusia mulai melakukan kegiatan penulisan pada berbagai benda. Benda itu dapat diteruskan dari satu generasi ke generasi berikutnya ataupun dapat dibaca oleh suku lain.

Berdasarkan bukti arkeologis diketahui bahwa perpustakaan pada awal mulanya tidak lain berupa kumpulan catatan transaksi niaga. Dengan kata lain, perpustakaan purba tidak lain merupakan sebuah kemudahan untuk menyimpan catatan niaga. Karena kegiatan perpustakaan purba tidak lain menyimpan kegiatan niaga maka ada kemungkinan bahwa perpustakaan dan arsip semula bersumber pada kegiatan yang sama untuk kemudian terpisah.

Dari kegiatan itu, ternyata bahwa sejak semula salah satu kegiatan perpustakaan ialah menyimpan produk tulisan masyarakat sekaligus juga perpustakaan merupakan produk masyarakat karena tak ada perpustakaan tanpa ada masyarakat.

2.2.1.5 Teknologi Informasi Perpustakaan

Perpustakaan sebagai institusi pengelola informasi merupakan salah satu bidang yang tidak lepas dari pengaruh teknologi informasi yang telah berkembang pesat. Penerapan teknologi informasi di perpustakaan dapat difungsikan dalam berbagai bentuk seperti :

1. Teknologi informasi digunakan sebagai Sistem Informasi Manajemen Perpustakaan. bidang pekerjaan yang dapat diintegrasikan dengam sistem informasi adalah pengadaan, inventarisasi, katalogisasi, sirkulasi bahan pustaka, pengelolaan anggota, statistik dan sebagainya.
2. Teknologi informasi sebagai sarana untuk menyimpan, mendapatkan dan menyebarluaskan informasi dalam bentuk *digital*. Bentuk penerapan teknologi informasi dalam perpustakaan ini dikenal dengan Perpustakaan *Digital*.
3. Teknologi informasi sebagai alat untuk memberikan keamanan dan kenyamanan dalam perpustakaan. Melalui fasilitas semacam *gate keeper*, *security gate*, CCTV dan sebagainya.

Peran dari teknologi informasi adalah sebagai perangkat (*tools*) yang digunakan untuk otomatisasi kerja. Dengan kerja yang telah terotomatisasi maka banyak manfaat yang didapatkan dalam pengelolaan perpustakaan seperti:

1. Meningkatkan mutu layanan kepada pengguna perpustakaan
2. Mengefisiensikan pekerjaan dalam perpustakaan
3. Memberikan kemudahan dalam pengambilan keputusan
4. Pengembangan infrastruktur

Seiring perkembangan dan kebutuhan untuk otomatisasi kerja di perpustakaan, kini sudah banyak tersedia perangkat lunak otomasi perpustakaan baik dari luar maupun dari dalam negeri dengan berbagai keunggulan yang ditawarkan dan harga yang bervariasi. Software yang sudah dikenal antara lain CDS/ISIS, WINISIS yang dikembangkan oleh Unesco dan Senayan yang dikembangkan oleh Kemendiknas (Kementerian Pendidikan dan Kebudayaan Republik Indonesia) yang bersifat gratis (*freeware*) bahkan banyak dari perguruan tinggi yang membuat dan mengembangkan sistem perpustakaannya sendiri,

seperti SIPUS WEB di UGM (Universitas Gajah Mada), Sipis di IPB (Institut Pertanian Bogor), dan sebagainya.

Perkembangan terakhir menunjukan kecepatan pengembangan perpustakaan telah banyak dipengatuhi oleh sentuhan teknologi. Hal ini karena pemanfaatan teknologi mampu meningkatkan fungsi peran perpustakaan sebagai media penyebaran informasi dan ilmu pengetahuan serta dengan teknologi mampu meningkatkan kecepatan efektivitas kerja dari pengelola perpustakaan itu sendiri.

Teknologi saat ini sudah memungkinkan adanya *self-services* dalam layanan sirkulasi dalam perpustakaan melalui fasilitas *barcoding* dan RFID (*Radio Frequency Identification*), bahkan penerapan teknologi komunikasipun sudah mulai digunakan seperti penggunaan SMS, Faksimili dan Internet.

Pada awal perkembangannya, penerapan teknologi identifikasi koleksi perpustakaan sangat mengandalkan *barcode.* Walaupun terbukti murah dan dapat mengurangi kesalahan petugas (*human error*) dalam mengidentifikasi koleksi perpustakaan dan keanggotaan, *barcode* ini ternyata mempunyai banyak kelemahan yaitu selain karena hanya bisa dikenali dengan cara mendekatkan *barcode* tersebut ke sebuah pembaca langsung (*reader*), juga karena mempunyai kapasitas penyimpanan data yang sangat terbatas dan tidak bisa ditulis ulang sehingga menyulitkan untuk menyimpan ataupun memperbaharui data. Kekurangnan lain dari sistem ini adalah waktu yang dibutuhkan untuk mengenali setiap *barcode* dan ditambah jika *barcode* kotor atau terhalang akan menambah waktu lebih lama lagi, sehingga ini akan membuat antrian orang-orang yang ingin melakukan transaksi peminjaman dan pengembalian buku terutama untuk perpustakaan yang jumlah layanan perharinya cukup banyak.

Seiring dengan perkembangan ilmu pengetahuan dan teknologi serta tuntutan mutu layanan di perpustakaan, Saat ini sudah mulai dikembangkan RFID untuk digunakan sebagai perangkat identifikasi koleksi perpustakaan. RFID dikembangkan sebagai pengganti atau penerus teknologi *barcode*. Implementasi teknologi RFID di dunia perpustakaan, khususnya di Indonesia saat ini masih tergolong baru. Oleh karena itu, implementasi RFID di dunia perpustakaan akan memberikan nilai eksklusifitas dan dapat mewujudkan revolusi dalam manajemen

perpustakaan modern. RFID memberikan keunggulan yang signifikan bila dibandingkan dengan teknologi *barcode* dan *tag anti-theft* (anti-pencurian). Keunggulan utama ada pada mengungkatnya pelayanan serta penghematan biaya operasional pengelola perpustakaan.

2.2.2 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan [4]. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DMBS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

2.2.2.1 DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [4]. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

2.2.2.2 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS[1]. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL yang secara umum terdiri dari 4 hal sebagai berikut:

1. Memasukkan data (*insert*)

```
INSERT INTO Tabel_mahasiswa
(nim, nama, tanggal_lahir)
VALUES
('10109240', 'Fahmi', '1991-05-03');
```

Query di atas digunakan untuk memasukkan data mahasiswa dengan NIM 10109240, nama Fahmi, dan tanggal lahir 3 Mei 1991 ke tabel "Tabel_mahasiswa".

2. Mengubah data (*update*)

```
UPDATE Tabel_mahasiswa
SET
Tanggal_lahir='1990-01-01'
  WHERE
    Nim='10109240';
```

Query di atas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM = 10109240 menjadi 1 Januari 1990 dalam tabel "Tabel_mahasiswa".

3. Menghapus data (*delete*)

```
DELETE FROM Tabel_mahasiswa
  WHERE
    Nim='10109240';
```

Query di atas dugunakan untuk menghapus data mahasiswa dengan NIM =10109240 dari tabel "Tabel_mahasiswa".

4. Menampilkan data (*select*)

```
SELECT nim,nama
FROM
Tabel_mahasiswa
  WHERE
    Nim='10109240';
```

Query di atas dugunakan untuk menampilkan data mahasiswa yang terimpan dalam "Tabel_mahasiswa" dengan NIM = 10109240.

2.2.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [4]. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.

3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.3.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya) [4]. Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian

perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

   Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat *runtime* yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat *runtime* (saat sebuah

program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

   Metode atau operasi dapat berasal dari *event*, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya *Read, Write, Move, Copy,* dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

   Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.3.2 Pengenalan UML

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan

pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

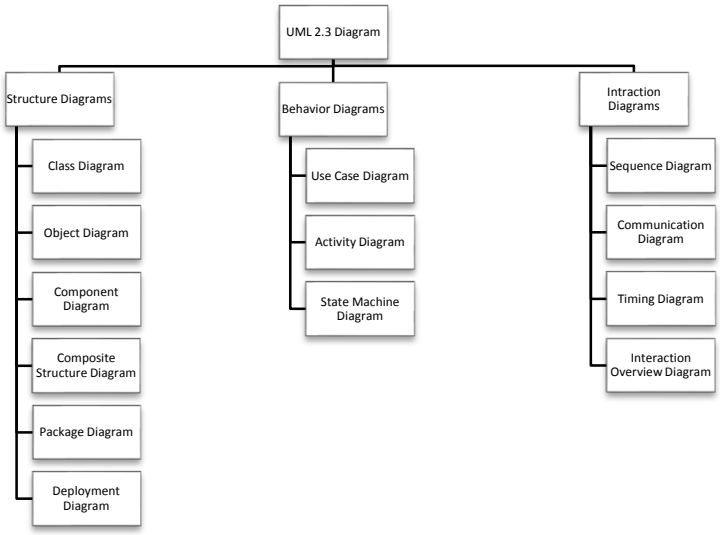
Pada perkembangan teknil pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan voisual untu menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan digaram dan teks-teks pendukung.

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

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

2.2.3.3 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.3.



Gambar 2.3 Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

2.2.3.4 Use Case Diagram

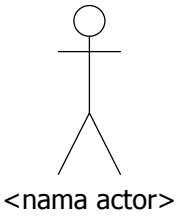
Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang disebut aktor dan *Use Case*.

2.2.3.4.1 Actor

Actor adalah sesuatu (entitas) yang berhubungan dengan sistem dan berpartisipasi dalam use case. Actor menggambarkan orang, sistem atau entitas eksternal yang secara khusus membangkitkan sistem dengan input atau masukan kejadian-kejadian, atau menerima sesuatu dari sistem. Actor dilukiskan dengan peran yang mereka mainkan dalam use case, seperti Staff, Kurir dan lain-lain.

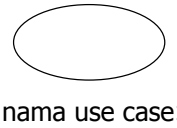
Dalam use case diagram terdapat satu aktor pemulai atau initiator actor yang membangkitkan rangsangan awal terhadap sistem, dan mungkin sejumlah aktor lain yang berpartisipasi atau participating actor. Akan sangat berguna untuk mengetahui siapa aktor pemulai tersebut.



Gambar 2.4 Bentuk Actor dalam UML

2.2.3.4.2 Use Case

Use case yang dibuat berdasarkan keperluan aktor merupakan gambaran dari "apa" yang dikerjakan oleh sistem, bukan "bagaimana" sistem mengerjakannya. Use case diberi nama yang menyatakan apa hal yang dicapai dari interaksinya dengan aktor.

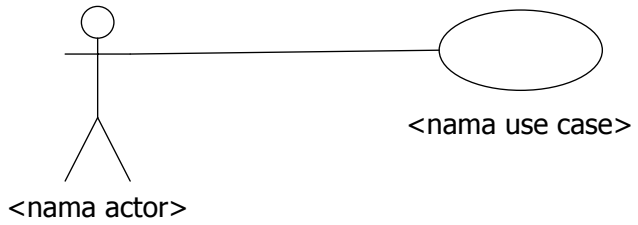


Gambar 2.5 Bentuk Use Case dalam UML

2.2.3.4.3 Relationship

Relasi (relationship) digambarkan sebagai bentuk garis antara dua simbol dalam use case diagram. Relasi antara actor dan use case disebut juga dengan asosiasi (association). Asosiasi ini digunakan untuk menggambarkan bagaimana hubungan antara keduanya.

Relasi-relasi yang terjadi pada use case diagram bisa antara actor dengan use case atau use case dengan use case.



Gambar 2.6 Bentuk Relationship dalam UML

Relasi antara use case dengan use case :

a. Include, pemanggilan use case oleh use case lain atau untuk menggambarkan suatu use case termasuk di dalam use case lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.
b. Extend, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak kontrol form dan mendeklarasikan ekstension pada use case utama. Atau dengan kata lain adalah perluasan dari use case lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis berpanah dengan tulisan <<extend>>.
c. Generalization/Inheritance, dibuat ketika ada sebuah kejadian yang lain sendiri atau perlakuan khusus dan merupakan pola berhubungan base-parent use case. Digambarkan dengan garis berpanah tertutup dari base use case ke parent use case.

2.2.3.5 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan

disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

2.2.3.6 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sequence yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence sehingga semakin banyak *use case* yang didefinisikan maka diagram sequence yang harus dibuat juga semakin banyak.

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.2.3.7 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan

variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas ebaiknya memiliki jenis-jenis kelas berikut :

1) Kelas Main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2) Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3) Kelas yang diambil dari pendefinisian *usecase*

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4) Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan coupling. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *choesion* yang kuat dan kadar *coupling* yang lemah.

Dalam class diagram terdapat beberapa relasi (hubungan antar class) yaitu:

1. Generalization dan Inheritence

Diperlukan untuk memperlihatkan hubungan pewarisan (inheritance) antar unsur dalam diagram kelas. Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari kelas yang berada dalam hirarki pewarisannya.

2. Associations

   Hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui ekstensi class lain. Dalam notasi UML kita mengenal asosiasi 2 arah (bidirectional) dan 1 arah (undirectional).

3. Aggregation

   Hubungan antar-class dimana class yang satu (part class) adalah bagian dari class lainnya (whole class).

4. Composition

   Aggregation dengan ikatan yang lebih kuat. Di dalam composite aggregation, siklus hidup part class sangat bergantung pada whole class sehingga bila objek instance dari whole class dihapus maka objek instance dari part calss juga akan terhapus.

5. Depedency

   Hubungan antar-class dimana sebuah class memiliki ketergantungan pada class lainnya tetapi tidak sebaliknya.

6. Realization

   Hubungan antar-class dimana sebuah class memiliki keharusan untuk mengikuti aturan yang ditetapkan class lainnya. Biasanya realization digunakan untk menspesifikasikan hubungan antara sebuah interface dengan class yang mengimplementasikan interface tersebut.

2.2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi, dipercaya dan bekerja secara efisien menggunakan mesin. Rekayasa perangkat lunak lebih fokus pada

praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat bagi pengguna. Rekayasa perangkat lunak fokus pada bagaimana membangun perangkat lunak yang memenuhi kriteria berikut:

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiiring berkembangnya teknologi dan lingkungan.
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi.
3. Efisien dari segi sumber daya dan penggunaan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan.

Pekerjaan yang terkait dengan rekayasa perangkat lunak dapat dikategorikan menjadi tiga buah kategori umum tanpa melihat area dari aplikasi, ukuran proyek atau kompleksitas perangkat lunak yang akan dibangun, yaitu:

a. Fase Pendefinisian (*Definition Phase*)

   Fase pendefinisian fokus pada "*what*" yang artinya harus mencari tahu atau mengidentifikasi informasi apa yang harus diproses, seperti apa fungsi dan performansi yang diinginkan, seperti apa prilaku sistem yang diinginkan, apa kriteria validasi yang dibutuhkan untuk mendefinisikan sistem.

b. Fase Pengembangan (*Development Phase*)

   Fase pengembangan fokus pada "*how*" yang artinya selama tahap pengembangan perangkat lunak seorang perekayasa perangkat lunak (*software engineer*) berusaha untuk mendefinisikan bagaimana data diinstrukturkan dan bagaimana fungsi-fungsi yang dibutuhkan diimplementasikan, bagaimana karakter antarmuka tampilan, bagaimana desanin ditranslasikan ke dalam bahasa pemrograman dan bagaimana pengujian dijalankan.

c. Fase Pendukung (*Support Phase*)

   Fase pendukung fokus pada perubahan yang terasosiasi pada perbaikan kesalahan (*error*), adaptasi yang dibutuhkan pada lingkungan perangkat lunak yang terlibat dan perbaikan yang terjadi akibat perubahan kebutuhan pengguna. Fase pendukung terdiri dari empat tipe perubahan antara lain:

1. Koreksi (*Correction*), yaitu pemeliharaan dengan melakukan perbaikan terhadap kecacatan perangkat lunak.
2. Adaptasi (*Adaptation*), yaitu merupakan tahap untuk memodifikasi perangkat lunak guna mengakomodasi perubahan lingkungan luar dimana perangkat lunak dijalankan.
3. Perbaikan (*Enchancement*), pemeliharaan perfektif atau penyempurnaan melakukan eksekusi atau penambahan pada kebutuhan fungsional sebelumnya.
4. Pencegahan (*Prevention*), pencegahan atau sering disebut juga dengan rekayasa ulang sistem (*software reengineering*) harus dikondisikan untuk mempu melayani kebutuhan yang diinginkan pemakainya.

2.2.5 Perangkat Lunak Pendukung

Adapun beberapa perangkat lunak pendukung dalam penyusunan Laporan Integrasi Teknologi RFID Pada Sistem Informasi Perpustakaan Untuk Pembangunan Perangkat Lunak Otomasi Perpustakaan di Institut Pemerintahan Dalam Negeri (IPDN) adalah

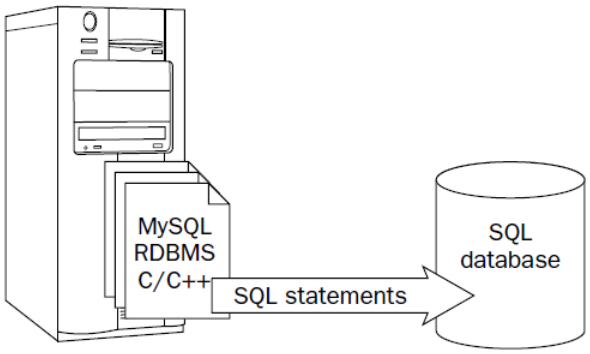
2.2.5.1 MYSQL

MySQL adalah sebuah produk *Relational Database Management System* (RDBMS). MySQL dapat mengakses data, memanipulasi data, melindungi data dari korupsi dan inkonsistensi serta mempertahankan metadata yang diperlukan untuk menentukan data yang disimpan dalam *database*. Perbedaan antara DBMS dengan RDBMS adalah bukan hanya dari mendukung atau tidaknya penyimpanan data dalam tabel, tetapi juga hubungan antartabel. MySQL menyediakan banyak fitur yang mendukung sebuah lingkungan yang aman untuk menyimpan, memelihara, dan mengakses data. Berikut kelebihan-kelebihan yang ditemukan di MySQL:

1. Skalabilitas, MSQL dapat menangani *database* yang besar, yang telah dibuktikan implementasinya dalam organisasi seperti Yahoo, Google, Cisco, HP, NASA dan lain sebagainya.

2. Portabilitas, MySQL dapat berjalan pada berbagai macam sistem operasi termasuk Windows, Unix, Linux, Solaris dan Mac OS. Juga dapat berjalan pada arsitektur yang berbeda, mulai dari *low-end* PC sampai *high-end mainframe*.
3. Konektivitas, MySQL sepenuhnya mendukung jaringan dan dapat diakses dari mana saja di internet serta pengguna dapat mengakses *database* MySQL secara bersamaan. MySQL juga menyediakan berbabagai macam API (*Aplication Program interface*) untuk mendukung konektivitas aplikasi yang ditulis dalam bahasa C, C++, Perl, PHP, Java, Phyton, C# dan lain sebagainya.
4. Keamanan, MySQL mencakup seluruh keamanan yang kuat untuk mengontrol akses data dan juga mendukung *Secure Socket Layer* (SSL) *Protocol*
5. Kecepatan, MySQL dikembangkan dengan kecepatan.
6. Mudah digunakan, MySQL mudah untuk digunakan dan diimplementasikan.
7. *Open Source*, MySQLAB membuat kode MySQL tersedia untuk digunakan setiap orang. filosofi *open source* memungkinkan khalayak global untuk berpartisipasi dalam pengembangan

Sebuah DBMS/RDBMS tidak dapat lepas dari SQL (*Structured Query Language*). SQL merupakan sebuah bahasa yang digunakan untuk mengelola dan berinteraksi dengan data dalam *database* relasional. SQL adalah bahasa *database* yang paling universal digunakan, dan itu telah menjadi bahasa standar untuk manajemen *database*. SQL bekerja sama dengan sebuah RDBMS untuk mendefinisikan struktur dari *database*, menyimpan data di *database* tersebut, memanipulasi data, mengambil data, mengontrol akses ke data, dan menjamin integritas data.



Gambar 2.7 Interaksi SQL dengan MySQL RDBMS

Beberapa perintah dasar SQL yang sering dipergunakan pada MySQL:

1. *Create Database,* perintah yang digunakan untuk membuat database baru.
   Sintaks : CREATE DATABASE DATABASE_NAME
2. *Drop Database,* perintah yang digunakan untuk menghapus database.
   Sintaks : DROP TABEL TABEL_NAME
3. *Create Tabel,* perintah yang digunakan untuk membuat tabel baru.
   Sintaks : Create Tabel tabel_name (create_definition)
4. *Describe,* perintah yang digunakan untuk mendeskripsikan tabel.
   Sintaks : Describe (Desc) tabel [colum]
5. *Alter Tabel*, perintah yang digunakan untuk memodifikasi tabel.
   Sintaks : Alter [Ignor] Tabel table_name
6. *Drop Tabel*, perintah yang digunakan untuk menghapus tabel.
   Sintaks : Drop Tabel tabel_name [tabel_name..]
7. *Delete*, perintah yang digunakan untuk menghapus record dri tabel.
   Sintaks : Delete From tabel_name Where Where_definiition
8. *Select*, perintah yang digunakan untuk query ke database.
   Sintaks : select * from tabel_name

MySQL mendukung beberapa API yang memungkinkan aplikasi yang ditulis dalam berbagai jenis bahasa pemrograman untuk berkomunikasi dengan database MySQL. Berikut beberapa API yang didukung oleh MySQL:

1. C API, adalah antarmuka pemrograman utama yang memungkinkan aplikasi C/C++ untuk menghubungkan ke MySQL. Sebagian besar

aplikasi klien termasuk dalam distribusi MySQL yang ditulis dalam C dan bergantung pada API ini.

2. ODBC, MySQL mendukung Open Database Connectivity (ODBC) melalui MySQL Connector / ODBC. ODBC adalah konektivitas database standar yang memungkinkan berbagai jenis aplikasi untuk menghubungkan ke berbagai jenis database.
3. JDBC, MySQL mendukung Java Database Connectivity (JDBC) melalui MySQL Connector / JDBC. JDBC adalah konektivitas *database* standar yang memungkinkan Java untuk menghubungkan ke berbagai jenis database.
4. PHP API, yang sekarang termasuk dengan *preprocessor* PHP, memungkinkan *script* PHP pada halaman Web untuk berkomunikasi secara langsung dengan *database* MySQL. Koneksi *database* dan permintaan data (melalui pernyataan SQL) dikodekan langsung dalam *script* PHP.

2.2.5.2 NET Framewok Bahasa Pemrograman C#

1. .NET Framework

Microsoft mengumumkan C# pada tahun 2000 bersamaan dengan pengumuman *platform* .NET Framework. .NET Framework adalah *platform* yang merupakan perwujudan teknologi modern. Framework .NET diciptakan untuk dapat memecahkan masalah yang banyak dihadapi dunia pemrograman secara lebih efisien. Metodologi perancangan berorientasi objek dan konsep software sebagai komponen tertanam dengan kuat di Framework.NET.

2. Bahasa Pemrograman C#

C# adalah bahasa pemrograman berbasis objek yang didukung oleh Microsoft .NET Framework. C# dibuat dengan tujuan untuk menyediakan bahasa pemrograman yang sederhana, modern, berorientasi objek, berpusat pada internet dan memiliki kinerja tinggi.[11] C# adalah salah satu dari banyak bahasa yang bisa dipakai untuk pemrograman .NET. Kelebihan utama bahasa ini adalah sintaksnya yang mirip C++, Java dan VB namun lebih sederhana dan mudah. Untuk

mencapai kemudahan inilah konsep-konsep sulit pada C++, Java dan VB disederhanakan dengan mencakup semua dukungan dukungan untuk pemrograman terstruktur, pemrograman berbasis komponen dan pemrograman berorientasi objek.

C# dikembangkan oleh sebuah tim yang dipimpin oleh Anders Hejlsbreg dan Scott Wiltamuth yang sebelumnya dikenal karena menciptakan Turbo Pascal dan merancang Borland Delphi.

Jantung dari setiap pemrograman berorientasi objek adalah adanya dukungan untuk mendefinisikan dan bekerja dengan kelas. Kelas mendefinisikan tipe baru yang memungkinkan programmer untuk memperpanjang bahasa. C# mengandung keyword untuk mendeklarasikan kelas baru, metode dan properti serta sebagai bahasa yang berorientasi objek C# juga dapat mengimplementasikan enkapsulasi, pewarisan dan polimorfisme.

C# sebagai bahasa pemrograman untuk Framework .NET memiliki ruang lingkup penggunaan yang sangat luas seperti pembuatan perangkat lunak dengan *user interface* Windows maupun *console* dan karena Framework .NET memberikan fasilitas untuk berinteraksi dengan kode yang *unmanaged*, *library* seperti DirectX dan OpenGL dapat digunakan di C#. C# juga dapat digunakan untuk pemrograman *web site*, *web service*, membangun perangkat lunak Zune, perangkat lunak permainan (desktop dan XBOX), perangkat lunak *mobile* dan lain sebagainya.

3. Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan pengembangan aplikasi, baik aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan *Library*. Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, C#, VB.NET, J++ dan J#.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *unmanaged*/*native code* (dalam bentuk bahasa mesin yang berjalan di atas

Windows) ataupun *managed* code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Beberapa keuntungan Visual Studio:

1. Dapat membuat aplikasi untuk beberapa platform termasuk Windows 7,Windows Server 2008 R2, SQL Server 2008, SharePoint. Office, Windows Azure, dan Windows Phone.
2. Lingkungan pemrograman dapat dipersonalisasi
3. Dukungan *Application Lifecycle Management* dapat memudahkan kolaborasi, melakukan *testing* ataupun *debugging*.

2.2.6 Radio Frequency Identification (RFID)

RFID (Radio Frequency Identification) adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau transponder (t*ag*) untuk menyimpan dan mengambil data jarak jauh. Label atau kartu RFID adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio.[4]

RFID merupakan sebuah teknologi yang menggunakan media radio frekuensi dalam proses identifikasinya. Identifikasi dilakukan secara otomatis terhadap objek-objek atau manusia tanpa memerlukan operasi manual. Maka dapat disimpulkan, RFID adalah teknologi penangkapan data yang dapat digunakan secara elektronik untuk mengidentifikasi, melacak dan menyimpan informasi yang tersimpan dalam t*ag* RFID. [6]

Teknologi RFID merupakan pengembangan dari teknologi identifikasi sebelumnya yaitu *barcode.* Keunggulan dari RFID yang amat signifikan jika dibandingkan dengan *barcode* adalah sebagai berikut:

1. Data yang dapat disimpan oleh *tag* RFID lebih banyak serta dapat dilakukan program ulang atas data yang tersimpan didalamnya.

2. *Barcode* harus di-*scan* secara optik, memerlukan kontak *line-of-sight* dengan *reader* sehingga peletakan fisik dari objek yang di*scan* harus tepat. Dan bahkan diperlukan campur tangan manusia untuk melakukan scanning terhadap *barcode.* Sedangkan *tag-tag* RFID dapat dibaca tanpa kontak *line-of-sight* dan tanpa penempatan yang presisi.

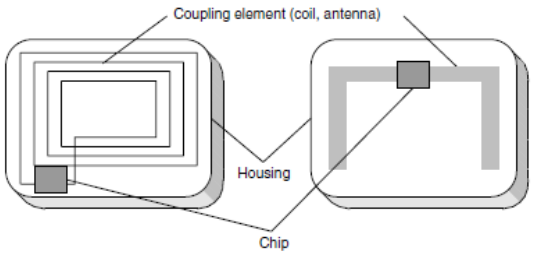
2.2.6.1 Komponen Utama RFID

RFID terdiri atas 3 komponen utama, yaitu:

1. RFID t*ag*, berisi antena sebagai *coupling element* yang memungkinkan untuk menerima dan merespon terhadap suatu query yang dipancarkan oleh suatu RFID *reader* . Juga terdapat chip yang mampu menyimpan sejumlah informasi unik.
2. *Reader* RFID, berisi modul frekuensi radio (*transmitter* dan *receiver),* pengontrol dan *coupling element* ke t*ag* .
3. *Host computer*, terdiri dari atas Basis data yang menyimpan semua data yang terkandung dalam t*ag.* Sistem computer yang mengatur alur informasi dari item-item yang terdeteksi dala:m lingkup sistem RFID dan mengatur komunikasi antara *tag* dan *reader* .

2.2.6.1.1 RFID Tag

RFID *tag* adalah sebuah benda kecil, komponen yang terdiri dari chip dan antenna. *Tag* ini dapat berupa stiker adesif yang ditempelkan pada suatu barang atau produk. Selain itu, *tag* ini juga dapat berupa koin dan kartu. Bentuk dan struktur dari *tag* ini dibuat fleksible sesuai dengan objek yang akan diidentifikasi. Gambar 2.8 menunjukkan *layout* dasar dari sebuah RFID *tag :*



Gambar 2.8 Layout Dasar RFID Tag [3]

RFID *tag* ini akan bekerja ketika menerima sinyal dari RFID *reader* dan akan memantulkan kembali sinyal tersebut. Ada 3 jenis RFID *tag* berdasarkan sifatnya, yaitu :

1. *Passive* RFID t*ag*

Passive RFID t*ag* adalah *tag* yang bersifat pasif karena tidak memiliki power supply sendiri. *Tag* ini bekerja dengan berbekal induksi listrik yang ada pada antenna kecil yang terkandug didalamnya. Bila ada frekuensi radio scanning yang masuk dari *reader* barulah induksi listrik akan terjadi sehingga t*ag* dapat merespon dengan mengirimkan data (informasi unik) yang disimpannya. Biasanya informasi sederhana seperti hanya nomor ID saja.

Dengan tidak adanya power supply pada t*ag* pasif maka ukurannya bisa dibuat sekecil mungkin. Saat ini RFID t*ag* yang sudah beredar di pasaran ada yang bisa diletakkan di bawah kulit. Namun jarak jangkauan dimana *tag* ini dapat terdeteksi oleh *reader* sangat terbatas (beberapa cm). Gambar 2.12 dibawah ini adalah contoh dari *Tag* pasif dengan spesifikasi sebagai berikut:

1) Frekuensi kerja : 125 kHz
2) Jarak pembacaan : 8-14 cm
3) Format : GK4001
4) Data (64-bit / dll) : 64-bit



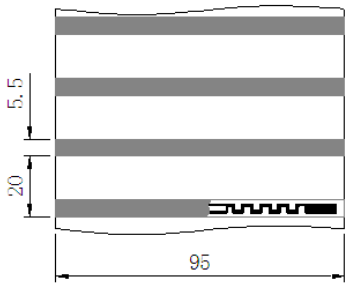
Gambar 2.9 RFID Tag Pasif

2. *Active* RFID *Tag*

Active RFID *tag* memiliki power supply sendiri, misalnya baterai yang dapat digunakan sebagian atau seluruhnya untuk mengaktifkan *circuit microchip* dan antenna, juga untuk mengirimkan sinyal ke *reader* . Beberapa *tag* juga dapat dihubungkan dengan catu daya dari luar. *Tag* ini memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung

berbagai macam informasi di dalamnya. Jarak jangkauan dari RFID *tag* yang aktif ini bisa sampai sekitar 10 meter dan dengan umur baterai yang bisa mencapai beberapa tahun lamanya. Dibawah ini adalah contoh dari *tag* aktif dengan spesifikasi sebagai berikut:

a. Karakteristik Fisik



Gambar 2.10 Karakrteristik Fisik RFID Tag Aktif

1) Ukuran : 95 × 5.5 × 0.28mm (3,7 × 0,22 × 0.01in)
2) Bahan Dasar : PET; alumunium
3) Bahan Casing : Perekat
4) Diameter Bagian Inti : 76.2 mm (3in)
5) Inti Lebar : 95 mm (3.7in)
6) Jumlah satu paket : 5.000 inlays per roll; sampai 12 gulungan per karton
7) Diameter Gulungan : 180 ± 5mm (7,1 ± 0.2in)
8) Berat : 2.32kg ± 0.05kg/roll; 10.0kg ± 0.2kg/carton
   (5,1 ± 0.11lbs/roll; 22,0 ± 0.44lbs/carton)

b. Spesifikasi Kinerja

1) Frekuensi Kerja : 860-960Mhz
2) Standar yang didukung : EPCglobal Kelas 1 Gen 2; ISO 18000-6C
3) Jarak Baca : ± 6 meter / 19.7 kaki (tergantung *reader*)
4) Jarak Tulis : ± 4 meter / 13.1 kaki (tergantung *reader*)
5) Jenis Chip : UCODE G2XM
6) Memori : 240-bit EPC; 64-bit TID;
   512-bit user memory
7) Fungsi : Read/Write
8) Data Retensi : 50 tahun

9) Daya Tahan Tulis : 100.000 siklus
10) Anti-collision : Operasi beberapa *Tag*



Gambar 2.11 RFID Tag Aktif XCTF-8102A

3. *Semipasif* RFID *Tag*

Tag semipasif menyerupai t*ag* aktif dimana *tag* jenis ini juga mempunyai catu daya sendiri, namun baterai yang dimiliki hanya digunakan untuk mengaktifkan *microchip* dan antenna, tidak digunakan untuk mengirimkan sinyal ke *reader* . Untuk proses *sinyal broadcast,* metodenya sama dengan cara *tag* pasif memantulkan sinyal ke *reader* .

Ketika mengaktifkan *tag,* catu daya yang dipakai adalah catu daya sendiri, sehingga tidak dipengaruhi oleh besarnya daya yang di induksikan oleh *reader* . Tetapi, saat pengiriman balik, daya yang digunakan sepenuhnya berasal dari *reader,* sehingga akan terjadi kegagalan kalau jaraknya terlalu jauh. Berikut adalah Tabel yang merupakan perbadingan spesifikasi beberapa jenis *tag* .

Tabel 2.1 Perbandingan beberapa jenis *Tag*

	Tag Pasif	*Tag* Semipasif	*Tag* Aktif
Catu daya	Eksternal (dari *reader*)	Baterai internal	Baterai internal
Rentang baca	Dapat mencapai 6 meter	Dapat mencapai 30 meter	Dapat mencapai 228 meter
Tipe Memori	Umumnya read-only	Read-write	Read-write
Harga	$0.2 hingga beberapa dollar	$2 hingga $10	$20 atau lebih
Usia *Tag*	Dapat mencapai 20 tahun	2 sampai 7 tahun	5 sampai 10 tahun

Berdasarkan tipe memori yang dimiliki oleh t*ag* , maka terdapat 3 variasi *tag :*

1. *Tag read-only, tag* ini memiliki kapasitas memori yang kecil (biasanya kurang ari 64 bit) dan data yang terprogram didalamnya bersifat permanen sehingga data tidak dapat diubah. *Tag* ini hanya bisa digunakan sekali, tidak dapat re-use. Informasi yang terkandung didalam *tag* seperti ini terutama adalah informasi identifikasi item. *Tag* pasif biasanya memiliki tipe memori seperti ini.
2. *Tag read/write,* data dapat di update jika diperlukan. Namun sebagai konsekuensinya kapasitas memorinya akan lebih besar dan harganya lebih mahal dibandingkan t*ag read-only. Tag* seperti ini biasanya digunakan bila data yang tersimpan didalamnya perlu untuk di di update seiring dengan daur hidup produk, misalnya di pabrik.
3. *Tag write-once read many,* informasi disimpan sekali dan tidak bisa dilakukan perubahan berikutnya terhadap data tapi dapat dibaca berulang-ulang. *Tag* tipe ini memiliki fitur keamanan *read-only* dengan menambahkan fungsionalitas tambahan dari *Tag read/write.*

2.2.6.1.2 RFID Reader

Prinsip kerja RFID *reader* serupa dengan *tranceiver* radio, yaitu memancarkan dan menerima. *Reader* ini dalam kondisi siaga akan memancarkan gelombang elektromagnetik sesuai dengan daya jangkaunya. Ketika ada *tag* memasuki area jangkauannnya, t*ag* akan mendapat daya dari gelombang elektromagnet *reader* .

Dari daya yang diperoleh, *tag* memancarkan data yang dibawa. Data pancaran tersebut akan diterima oleh *reader* . Selanjutnya data yang diterima tadi akan diteruskan pada aplikasi untuk diolah sesuai dengan rancangan sistem.

Dalam fungsinya, RFID *reader* dituntut untuk dapat melakukan dua tugas, yaitu berkomunikasi melalui gelombang radio dan membaca data yang dibawa *tag* kemudian diteruskan ke aplikasi.



Gambar 2.12 RFIDReader RapidACCESS RFD220P

Gambar diatas adalah jenis *reader* RFID RapidACCESS RFD220P, RapidACCESS RFD220P adalah pembaca tetap / desktop terintegrasi yang dapat membaca dan menulis label elektronik atau yang dilengkapi dengan t*ag* ISO-18000-6B standar, EPC Kelas 1 standar G2 dan standar EM-4222. Berkisar membaca maksimal hingga 4 meter (tergantung t*ag*). Spesifikasinya yaitu sebagai berikut:

1. Read-Write *tag* : *tag* elektronik dengan ISO-18000-6B, EPC Class1 G2 danstandar EM-4222 .
2. Mendukung komunikasi data melalui port RS232, Wiegand pelabuhan, pelabuhan RS485
3. Modus kerja: Frekuensi Hopping Spread Spectrum (FHSS)
4. Max. RF output daya: 20 ~ 30dBm
5. Kecepatan membaca: merata, 6ms / 32bits.
6. Menulis kecepatan: merata, 50ms/32bits.
7. Mekanisme keamanan: Verifikasi keabsahan *Tag* .
8. Built-in anti-tabrakan: Baca hingga 100 *Tag* pada suatu waktu.
9. 3 (tiga) Mode Operasi Tersedia:
    a. CONTINOUS: *Tag* otomatis membaca secara berkala
    b. TRIGGER: Kontrol membaca *Tag* dengan memasukkan memicu sinyal
    c. COMMAND: Memulai membaca / menulis dengan perintah software / API

10. Power supply: 110V ~ 220V eksternal VAC auto mulai.
11. Dimensi: 250mm x 250mm x 77mm
12. Berat: 0.9kg (tanpa kabel dan bracket)
13. Suhu operasi: -20 C sampai 55 C
14. Jarak membaca *Tag* hingga 4 meter, menulis *Tag* hingga 1,5 meter (tergantung *Tag*)

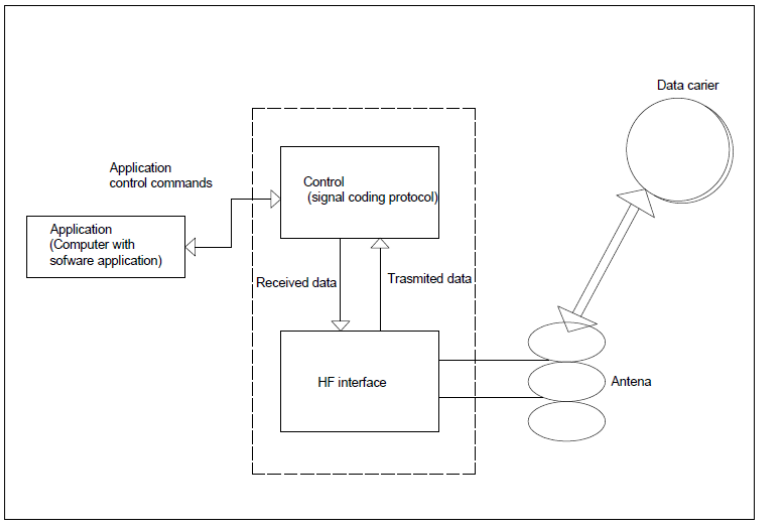


Gambar 2.13 RFID Reader ID-12 Innovations

Gambar diatas adalah salah satu jenis *reader* RFID jenis ID-12 Innovations. RFID tersebut merupakan sarana pengembangan RFID berbasis *reader* tipe ID-12 yang telah dilengkapi jalur komunikasi RS-232sehingga dapat dihubungkan ke COM port komputer secara langsung. Modul ini cocok untuk mengembangkan aplikasi-aplikasi berbasis RFID, misalnya: mesin absensi RFID, RFID access controller, dsb. Spesifikasinya adalah sebagai berikut:

1) Catu daya : 5VDC
2) Frekuensi kerja : 125 kHz
3) Jarak pembacaan : ± 12 cm
4) Antarmuka : UART TTL / RS-232), Wiegand26, maupun Magnetic ABA Track 2
5) Format kartu : EM4001
6) Fitur : Modul ID-12 dapat diganti dengan modul ID-Series yang lain.
7) Dimensi : 8.1 cm (p) x 6.2 cm (l) x 1.8 cm (t)

Reader dibedakan berdasarkan kapasitas penyimpanannya, kemampuan pemrosesannya, serta frekuensi yang dapat dibacanya. Kompabilitas *reader* terhadap t*ag* juga perlu diperhatikan karena tidak semua t*ag* dengan *reader* akan *match*, ada pasangannya masing-masing. Berikut ini gambar 2.14 adalah bagan 2 blok fungi dari *reader*



Gambar 2.14 Bagan 2 Blok Fungsi dari Reader [3]

a. HF Interface

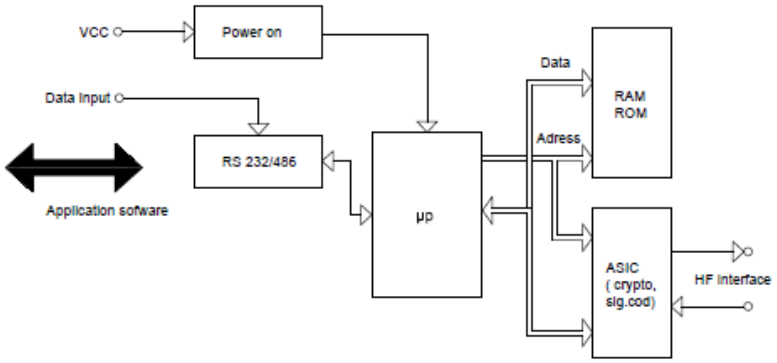
High Frequency Interface digunakan oleh *reader* untuk menghasilkan sinyal transmisi agar dapat mengaktifkan *tag* dan menyuplai daya (pada *tag* pasif), memodulasi sinyal transmisi untuk mengirimkan data ke *tag* , serta menerima dan mendemodulasikan sinyal transmisi dari sebuah *tag* .

HF Interface terdiri dari dua jalur sinyal yang terpisah, berdasarkan pada dua arah aliran data dari dan ke *tag* , Data yang dikirimkan ke *tag* melalui bagian transmisi dan data yang diterima dari t*ag* akan di proses pada bagian *receiver.*

b. *Control Sistem*

Gambar 2.15 menunjukkan bagian *control* sistem dari sebuah *reader* . *Control* sistem ini memiliki fungsi-fungsi sebagai berikut:

1. Berkomunikasi dengan aplikasi perangkat lunak dan melakukan perintah dari aplikasi perangkat lunak.
2. Mengontrol metode komunikasi dengan *tag*
3. Melakukan proses *coding* dan *decoding* dari sinyal informasi



Gambar 2.15 Diagram blok dari conrol unit [3]

Control unit memiliki sistem sesuai dengan mikroprosessor yang digunakan. Pertukaran data antara aplikasi perangkat lunak dan *reader* dilakukan menggunakan komunikasi serial melalui RS232 *interface*.

2.2.6.2 Sistem Pada Beberapa Kategori RFID

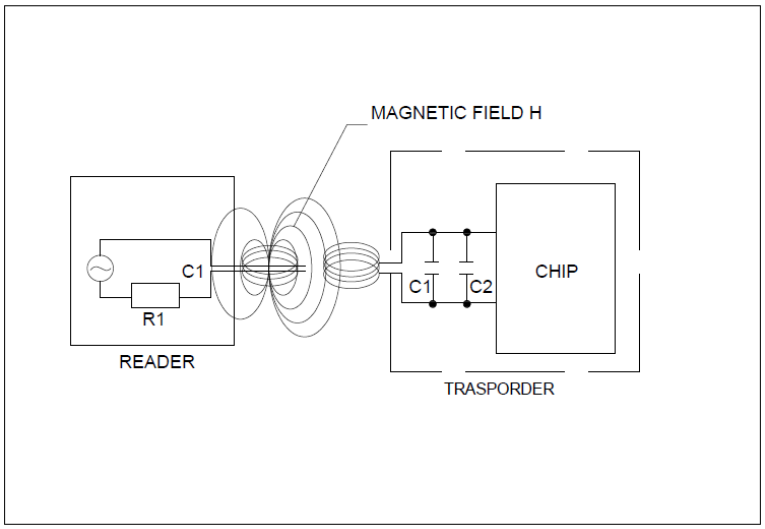
Sistem pada RFID dibagi menjadi 3 kategori berdasarkan sifat *tag* yang digunakan, yaitu sistem pada RFID pasif, sistem pada RFID aktif dan sistem pada RFID semipasif.

2.2.6.2.1 Sistem Pada RFID Pasif

Secara umum, proses yang terjadi saat sistem RFID pasif beroperasi adalah:

1. *Tag* memasuki medan elektromagnetik frekuensi radio
2. Sinyal RF mensuplai energi ke t*ag*
3. *Tag* mentransmisikan ID dan data yang tersimpan didalamnya
4. *Reader* menangkap data yang dikirim oleh *tag*
5. *Reader* mengirim data dari *tag* tersebut ke komputer
6. Komputer mengolah dan menganalisis data
7. Komputer memberikan instruksi kepada *reader*
8. *Reader* mentransmisikan data ke *tag*

Gambar 2.16 adalah ilustrasi dari proses yang terjadi pada sistem RFID pasif:



Gambar 2.16 Proses Catu Daya pada Sistem RFID Pasif [3]

Inductive coupled tag yang beroperasi secara pasif, terdiri dari *microchip* tunggal dan area kumparan yang luas sebagai antena. *Reader* yang menjadi pensuplai daya untuk mengaktifkan microchip, dimana antena *reader* akan membangkitkan medan elektromagnetik (dengan frekuensi tinggi) yang kuat dan akan menembus area kumparan dan di sekitar kumparan.

Sebagian medan akan menembus kumparan *tag* yang letaknya berjauhan dengan *reader* dan menghasilkan tegangan pada antena *tag* akibat proses induksi. Tegangan ini akan diserahkan oleh dioda pada rangkaian *tag* dan kemudian digunakan sebagai sumber tegangan untuk mengkatifkan chip.

Kapasitor dihubungkan secara paralel dengan kumparan antena untuk menghasilkan frekuensi resonansi yang sesuai dengan frekuensi *reader*. Kumparan antena dan kapasitor ini telah disesuaikan dengan frekuensi *reader*, dan tegangan induksi akan mencapai titik maksimum saat resonansi meningkat pada rangkaian paralel tersebut.

Pada *inductive coupled sistem* proses transmisi sinyal antara kedua perangkat tersebut tergantung pada kumparan yang dimiliki pada antena keduanya. Sinyal *feedback* dari *tag* ke antena *reader* dapat direpresentasikan sebagai impedansi pada kumparan antena *reader* . Perubahan resistor beban dari *on* dan *off* akan mengubah impedansi ini dan juga mengubah tegangan pada antena *reader* .

Perubahan resistor ini diatur oleh data, sehingga data dapat dikirimkan dari t*ag* ke *reader* , pengiriman ini disebut *load modulation.* Sinyal yang diterima oleh *reader* akan dimodulasi dengan cara menyerahkan tegangan yang masuk ke *reader.* Sehingga hasil modulasinya berupa amplitudo dari sinyal modulasi.

2.2.6.2.2 Sistem pada RFID semi pasif

Sistem yang ketiga adalah penggabungan sifat dan atribut dari sistem RFID pasif dan aktif. Sistem RFID semipasif ini mempunyai arsitektur yang cukup unik karena menggunakan baterai sendiri untuk sumber power dalam memproses data internal sedangkan untuk meradiasikan data kembali ke *reader* menggunakan power dari sinyal yang diterima dari *reader* .

2.2.7 Integrasi

Sistem integrasi (*integrated system*) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem-sistem komputerisasi dan software aplikasi baik secara fisik maupun secara fungsional. Sistem integrasi akan menggabungkan komponen sub-sub sistem ke dalam satu sistem dan menjamin fungsi-fungsi dari sub sistem tersebut sebagai satu kesatuan sistem.

Tantangannya adalah bagaimana merancang sebuah mekanisme mengintegrasikan sistem-sistem tersebut dengan *effort* paling minimal – bahkan jika diperlukan, tidak harus melakukan *refactoring* atau *re-developing* lagi sistem-sistem yang sudah ada.Metode yang dapat dipergunakan dalam membangun sistem terintegrasi yaitu :

1. ***Vertical Integration,*** merupakan proses mengintegrasikan sub-sub sistem berdasarkan fungsionalitas dengan menghubungkan sub-sub sistem yang sudah ada tersebut supaya bisa berinteraksi dengan sistem terpusat dengan tetap berpijak pada arsitektur sub sistem yang lama.
    a. Keuntungan yaitu dapat dilakukan dengan cepat dan hanya melibatkan beberapa entitas d*evelopment* yang terkait dalam proses pembuatan sistem lama.
    b. Kelemahan, metode ini tidak memungkinkan untuk mengimplementasikan fungsi-fungsi baru atau proses bisnis baru ke

dalam sub-sistem yang sudah ada – karena *effort* lebih tinggi ada di proses *"mempelajari"* arsitektur sistem lama dan menjadikannya acuan untuk membuat sistem terintegrasi

2. ***Star Integration (****spaghetti integration),* adalah proses mengintegrasikan sistem dengan cara menghubungkan satu sub sistem ke semua sub-sub sistem lainnya. Sebuah fungsi bisnis yang diimplementasikan dalam sebuah sub sistem akan di-*broadcast* ke semua sub-sub sistem lain yang dependen terhadap fungsi bisnis tersebut supaya dapat dipergunakan sebagaimana mestinya. Untuk integrasi sistem dengan ruang lingkup kecil atau menengah dan dengan pemisahan fungsi bisnis yang jelas dan spesifik, metode integrasi ini layak untuk dipertimbangkan
3. ***Horizontal Integration(****Enterprise Service Bus (ESB))*, merupakan sebuah metode yang mengintegrasikan sistem dengan cara membuat suatu *layer* khusus yang berfungsi sebagai *interpreter*, dimana semua sub-sub sistem yang sudah ada akan berkomunikasi ke *layer* tersebut. Model ini lebih menawarkan fleksibilitas dan menghemat biaya integrasi.

ESB memiliki banyak kelebihan jika diadopsi dalam merancang arsitektur sistem terintegrasi, yaitu antara lain :

1. Lebih cepat dalam melakukan penyesuaian dengan sistem yang telah ada
2. Meningkatkan fleksibilitas, mudah untuk diperbaharui mengikuti perubahan keperluan sistem *(system requirements)*
3. Membuat standar sistem sehingga bisa diaplikasikan di sub sistem mana pun
4. Porsi pekerjaan software development lebih banyak di *"konfigurasi"* daripada *"menulis code"* untuk integrasi
5. Dapat diterapkan mulai ruang lingkup kecil hingga di level *enterprise*

Namun metode *(ESB)* yang tampaknya ideal ini bukan berarti tidak ada kelemahan. Beberapa kelemahan yang cukup signifikan pengaruhnya antara lain :

1. Pembuatan standar sistem dalam Enterprise Message Model banyak berkutat di aspek analisis dan manajerial, biaya analisis benar-benar

tinggi karena perlu berkolaborasi dengan analis-analis yang bertanggung jawab terhadap arsitektur dan desain sistem-sistem yang telah ada.

2. Secara khusus memerlukan perangkat keras *(hardware)* yang spesifik, seperti misalnya *business-logic-server* yang independen dan tidak integral dengan salah satu atau sebagian dari sub sistem yang telah ada.
3. Perlu tambahan tenaga (SDM) berupa *Middleware Analyst* yang akan mengkonfigurasi, merawat, dan mengoperasikan layer *Enterprise Service Bus*.
4. Karena biasanya ESB mempergunakan XML sebagai bahasa komunikasi antar sistem, tentu akan memerlukan *resources* dan komputasi berlebih untuk melakukan *parsing-reparsing* dalam komunikasi data.
5. Memerlukan *effort* yang cukup tinggi dalam mengimplementasikan ESB karena cukup banyak *layer*/tingkatan aplikasi yang harus ditangani, tidak hanya aplikasi-aplikasi interface dari sub-sub sistem saja, melainkan juga *layer interpreter* yang juga memiliki karakteristik sebagai aplikasi juga.

Integrasi sistem informasi dapat bersifat hirarkis yaitu pada tingkat transaksi akan memberikan masukan data kepada sistem tingkat manajerial atau sering pula dalam arah sebaliknya. Interaksi hirarkis adalah paling banyak diidentifikasikan dan diitegrasikan karena manajer mengetahui bahwa informasi harus diringkaskan menurut jalur hirarki disamping sistem yang bersangkutan ada di bawah satu garis komando dan karena manajer dalam bidang fungsional akan lebih banyak mengetahui data apa yang ada dalam sistemnya. Keuntungan dari integrasi :

1. Membaiknya arus informasi di dalam sebuah organisasi.
2. Mendorong manajer untuk membagikan informasi yang dihasilkan oleh departemennya agar secara rutin mengalir ke sistem yang lain yang memerlukan.