BAB 2
TINJAUAN PUSTAKA

2.1 Ruang Lingkup Objek Penelitian

Objek penelitian yang sedang diteliti adalah Sekolah Menengah Kejuruan (SMK), yaitu SMKN 1 Katapang yang bertempat di JL. Ceuri Terusan Kopo Km. 13.5, Katapang, Kec. Katapang, Kab. Bandung Prop. Jawa Barat.

2.1.1 Sejarah Instansi

SMKN 1 Katapang berdiri Tahun 1999 dengan nama SMKN 4 Soreang. Unit Gedung Baru (UGB) SMKN 1 Katapang dibangun dari dana proyek LOAN OCF yang merupakan bantuan Pemerintah Jepang. Tahun Pelajaran 1999/2000 mulai menerima siswa baru untuk Program Keahlian Teknologi Penyempurnaan Tekstil, Teknik Elektro dan Mesin Perkakas (pendaftar ada 150 dan yang diterima 144 orang siswa), pada waktu itu KBM dilaksanakan di SMPN 1 Katapang. Tahun Pelajaran 2000/2001 SMKN 4 Soreang mulai menempati Unit Gedung Baru yang berlokasi di Jalan Ceuri Terusan Kopo KM 13,5 Desa Katapang, Kecamatan Katapang Kabupaten Bandung.

Pada waktu peresmian UGB SMKN 1 Katapang yang dihadiri oleh pejabat dari Dinas Pendidikan Provinsi dan Kabupaten, Pemerintah setempat dan tokoh masyarakat setempat pada tahun 2000, disepakati nama SMKN 4 Soreang menjadi SMKN 4 Katapang atas keinginan dari tokoh masyarakat bahwa nama SMKN 4 Soreang kurang sesuai karena berada di Kecamatan Katapang. Akhir Tahun 2000 keluar Surat Keputusan pembukaan Sekolah yang menetapkan nama sekolah yang dulu bernama SMKN 4 Soreang secara resmi adalah SMKN 1 Katapang Kabupaten Bandung.

2.1.2 Visi

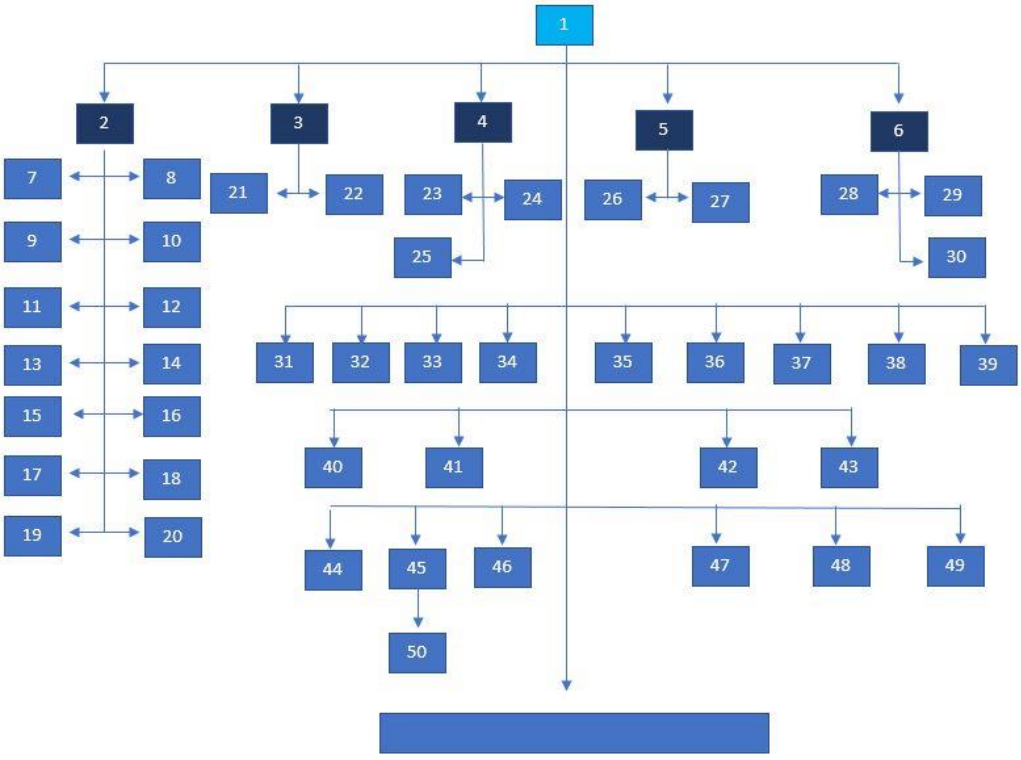
Menjadi pusat penyiapan generasi yang unggul, berkarakter kebangsaan, kompetitif dan *adaptable*

2.1.3 Misi

1. Menerapkan manajemen pelayanan yang responsif dan akuntabel.
2. Melaksanakan Pembelajaran inovatif dan profesional yang didukung oleh teknologi berdasarkan pada kearifan kebangsaan.
3. Meningkatkan sumber daya manusia (SDM) yang sesuai dengan perkembangan zaman berdasarkan kearifan kebangsaan.
4. Menghasilkan lulusan yang berkarakter unggul, kompetitif sebagai pemimpin di masa depan.

2.1.4 Struktur Organisasi

Struktur organisasi SMKN 1 Katapang dapat dilihat pada gambar 2.1



Gambar 2. 1 Struktur Organisasi SMKN 1 Katapang

Keterangan:

1. Dra. Hj. Adah Rodiyah, M.MPd. : Kepala Sekolah
2. H. Purwo Cahyono Hadi, S.ST. : Waka. Bidang Kesiswaan
3. H. Milad Doso Ismoyo, S.Pd., ST. M.MPd. : Waka Hubin
4. Edya Muhsin S.Pd., M.Pd. : Waka Kurikulum
5. Sumarna S.Pd. MM. : Waka Bidang Sarana dan Prasarana
6. Tatat Sudrajat, S.Pd. : Wakil Manajemen Mutu
7. Subiyono S.Pd., S.ST : Pembina Osis
8. Andri Aljani, S.Kom. : Koordinator Tim Kedisiplinan
9. Angga Perdana PS. S.Pd. : Pembina PASUS
10. Zein Fuad M, S.Pd. : Staf Kedisiplinan
11. Angga Yogagiri, SPd. : Pembina KIR
12. Lulu Unadid S., S.Pd.I : Pembina Kerohanian
13. Luki Setia, S.Pd. : Staf Kedisiplinan
14. Hj. Mike Wilasmi, S.Pd. : Staf Kedisiplinan
15. Asep Suratno, ST. : Pembina PLH
16. Sutarti S.Pd. MM. : Pembina Keputrian
17. Osa Pauliza, S.Pd. : Pembina Olah Raga
18. Fitriyah Fajar S.Pd. : Pembina Keputrian
19. Nira Sunarni, S. Sn : Pembina Kesenian
20. Nur Asyiah, S.Pd. : Pembina Keputrian
21. Purwo Trenggono S.Pd., MM. : Koordinator Prakerin
22. Tetty Sugiarti S.Pd, M.MPd. : Staf Hubin Bidang BKK
23. Empep Mustopa, S.Pd. : Pelaksana Pembelajaran
24. Tania Narulita, AT. MM. : Staf Kurikulum
25. Cecep Fitra G. : Staf Kurikulum
26. Yuliani Sutaryanti, S.Pd. MM. : Staf Sarana

27. Eka Natakusuma, S.Pd. : Staf Sarana
28. Evy Yeviana, S.Pd., MPKim : Staf WMM
29. Teti Rosita, ST. M.Mpd. : Staf WMM Bidang Pengembangan Dan Evaluasi
30. H. Elan Herlan, S.Pd. : Staf WMM Bidang Dokumen
31. Mamat Rahmat, S.Pd., S.ST. MM. : Ka. Prog. Elektronika Industri
32. Drs. Suwandi : Kepala Paket Keahlian TKR
33. Drs. Winarko : Kepala Paket Keahlian Mesin
34. Kartini S.SIT. M.A.P. : Kepala Paket Keahlian Tekstil
35. Hendi Hadiansah, S. Pd. S.ST. : Kepala Program TKI
36. Angga Aditia, ST. : Kepala Paket Keahlian RPL
37. Insan Yuliardi MS. S.KOM : Kepala Paket Keahlian Multimedia
38. JS. Budi Haryono, S.Pd., S.ST. M.Mpd. : Kepala Paket Keahlian Mekatronika
39. Panca Hudaya Sair, S.Pd. : Kepala Program TGM
40. Dra. Cacih Julaeha : Ka. Lab. Kimia
41. Neni Suryani S.Pd. M.Pfis : Ka. Lab Fisika
42. Sri Rusmiati S.Pd., MM. : Ka. Lab Bahasa
43. Nunung Sukmawati, S.Pd. MM. : Ka. Perpustakaan
44. Ennie Soesilaningsih S.Pd. MM. : Ka. Program Adaptif
45. Drs. Hj. Sumarni, MM : Ka. Program Normatif
46. Tri Indarti, S.Pd. : Koordinator BK
47. Drs. H. Usman Suherman : Pembina UKS
48. Andi Lesmana, S.Pd. : Pembina Pramuka Putera
49. Risma Fidiyanti S.Pd. : Pembina Pramuka Puteri

50. Dra. Sri Purwaningsih, MM. : Sekretaris Unit Produksi

2.2 Landasan Teori

Landasan teori merupakan kumpulan dari beberapa definisi dan konsep yang disusun secara sistematis mengenai variabel-variabel dalam penelitian. Landasan teori ini akan menjadi acuan dasar penulis dalam melakukan penelitian melalui uraian-uraian secara teoritis.

2.2.1 Informasi

Berikut adalah definisi informasi dari beberapa ahli [1]:

a. Informasi adalah data yang sudah dibentuk ke dalam sebuah formulir bentuk yang bermanfaat dan dapat digunakan manusia [11].
b. Informasi adalah penerangan, keterangan, pemberitahuan, kabar atau berita (tentang). Informasi juga adalah keterangan atau bahan nyata yang dapat disajikan dasar kajian analisis atau kesimpulan [12].
c. Informasi adalah data yang telah diproses atau diolah kedalam bentuk yang sangat berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya [13].
d. Informasi terdiri atas data yang telah didapatkan, diolah atau diproses atau sebaliknya yang digunakan untuk tujuan penjelasan, penerangan, uraian atau sebagai sebuah dasar untuk pembuatan ramalan atau pembuatan keputusan [14].

2.2.2 Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer yang saling terhubung. Tujuan utama dari jaringan komputer ini yaitu untuk pertukaran data dan informasi antar komputer dalam satu jaringan.

Keuntungan yang didapat dengan menggunakan jaringan komputer diantaranya:

a. Dapat mengakses data di komputer lain dari komputer yang anda gunakan.
b. Data yang digunakan dapat disimpan atau disalin ke beberapa komputer, sehingga bila salah satu komputer rusak maka salinan di komputer lain dapat digunakan.
c. Penggunaan printer, *scanner*, CD/DVD ROM dan perangkat lainnya dapat digunakan bersama-sama dengan pengguna lain.
d. Administrator jaringan dapat mengontrol data-data penting agar dapat diakses oleh pengguna yang berhak saja.
e. Penghematan biaya dapat dilakukan, karena sebuah perangkat dapat digunakan bersama-sama.

Berdasarkan jangkauan area atau lokasi, jaringan dibedakan menjadi 3 jenis yaitu:

a. *Local Area Network* (LAN)

   Local Area Network (LAN) merupakan jaringan yang menghubungkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. LAN dapat menggunakan media komunikasi seperti kabel dan *wireless*.

b. *Wide Area Network* (WAN)

   Wide Area Network (WAN) merupakan jaringan antara LAN satu dengan lain yang dipisahkan oleh lokasi yang cukup jauh. Contoh penggunaan WAN adalah hubungan antara kantor pusat dengan kantor cabang yang ada di daerah-daerah.

c. *Metropolitan Area Network* (MAN)

   Metropolitan Area Network (MAN) merupakan jaringan yang lebih besar dari jaringan LAN tetapi lebih kecil dari jaringan WAN. Jaringan MAN dan WAN sama-sama menghubungkan beberapa LAN yang membedakan hanya lingkup areanya yang berbeda.

Berdasarkan fungsinya jaringan komputer terbagi menjadi dua jenis yaitu:

a. Jaringan *Client Server*

Jaringan *client server* menghubungkan komputer server dengan komputer klien. Komputer server adalah komputer yang menyediakan fasilitas bagi komputer-komputer klien yang terhubung dalam jaringan. Sedangkan komputer klien adalah komputer yang menggunakan fasilitas yang disediakan komputer server. Komputer server pada sebuah jaringan client server disebut sebagai *Dedicated Server*, karena komputer yang digunakan hanya sebagai penyedia fasilitas komputer klien. Komputer server tidak dapat berperan sebagai komputer klien.

Keunggulan menggunakan jaringan *client server* yaitu:

1. Terdapat administrator jaringan yang mengelola sistem keamanan dan administrasi jaringan.
2. Komputer server difungsikan sebagai pusat data.
3. Pengaksesan data lebih tinggi karena penyediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer server.
4. Sistem *backup* data lebih baik, karena *backup* data dapat dilakukan terpusat di komputer server.

Kelemahan menggunakan jaringan *client server* yaitu:

1. Biaya mahal, karena membutuhkan komputer yang memiliki kemampuan tinggi yang difungsikan sebagai komputer server.
2. Kelancaran jaringan bergantung pada komputer server, bila komputer server mengalami gangguan maka jaringan akan terganggu.

b. Jaringan *Peer To Peer*

Jaringan *Peer To Peer* menghubungkan beberapa komputer dalam sebuah jaringan. Pertukaran data dapat dilakukan antar komputer yang terhubung tanpa perantara komputer server. Masing-masing komputer dapat berperan sebagai komputer server sekaligus sebagai komputer klien.

Keunggulan menggunakan jaringan *peer to peer* yaitu:

1. Semua komputer yang terhubung memiliki hak yang sama.

2. Biaya lebih murah karena tidak memerlukan adanya sebuah komputer server.
3. Kelancaran jaringan tidak bergantung pada komputer server.

Kelemahan menggunakan jaringan *peer to peer* yaitu:

1. Troubleshooting jaringan lebih rumit
2. Sistem keamanan jaringan ditentukan oleh masing-masing pengguna dengan mengatur keamanan pada fasilitas yang dimiliki.
3. *Backup* data dilakukan pada masing-masing komputer [2].

2.2.3 Intranet

Intranet adalah konsep LAN yang mengadopsi teknologi Internet dan mulai diperkenalkan pada akhir tahun 1995. Dengan kata lain intranet adalah LAN yang menggunakan standar komunikasi dan segala fasilitas internet. Intranet dapat diibaratkan berinternet dalam lingkungan local dan pada umumnya terkoneksi ke internet sehingga memungkinkan pertukaran informasi dan data dengan jaringan intranet lainnya melalui *backbone* Internet.

Intranet adalah sebuah jaringan privat (*private network*) yang menggunakan protokol-protokol Internet (TCP/IP) untuk membagi informasi rahasia perusahaanatau operasi dalam perusahaan tersebut kepada karyawannya. Kadang-kadang, istilah intranet hanya merujuk kepada layanan yang terlihat, yakni situs web internal perusahaan. Untuk membangun sebuah intranet maka sebuah jaringan harus memiliki beberapa komponen yang membangun Internet yaitu protokol Internet (Protokol TCP/IP, alamat IP, dan protokol lainnya), *client* dan juga *server*. Protokol HTTP dan beberapa protokol Internet lainnya (FTP, POP3, atau SMTP) umumnya merupakan komponen protokol yang sering digunakan.

Umumnya, sebuah intranet dapat dipahami sebagai sebuah versi pribadi dari jaringan internet atau sebagai sebuah versi dari Internet yang dimiliki oleh sebuah organisasi. Adapun manfaat dari Intranet adalah:

1. Membuat perusahaan atau institusi menjadi lebih hemat dan efisien, karena

sistem informasi berubah dari cara konvensional ke cara yang memanfaatkan web dan *database*.

2. Perolehan informasi yang cepat dan detail serta sistem manajemen yang canggih akan membuat perusahaan atau intsitusi tersebut memimpin dalam bidangnya.
3. Pertukaran data antar lokasi secara *point to point*, seperti *video conference*, transfer data atau file dan telepon intranet [3].

2.2.4 Pemodelan Analisis

Model analisis merupakan representasi teknis yang pertama dari sistem. Pertama analisis terstruktur adalah metode pemodelan klasik, dan analisis berorientasi objek.

Analisis terstruktur merupakan salah satu teknis analisis yang menggunakan pendekatan berorientasi fungsi. Teknik ini mempunyai sekumpulan perangkat yang memungkinkan analisis sistem mendefinisikan spesifikasi fungsional perangkat lunak secara terstruktur, yang dimaksud dengan perangkat analisis terstruktur adalah alat bantu pemodelan yang digunakan untuk menggambarkan hasil pelaksanaan analisis.

Entity-relational diagram (ERD) adalah notasi yang digunakan untuk melakukan aktivitas pemodelan data, sedangkan *data flow diagram* (DFD) memberikan informasi sebagai dasar bagi pemodelan fungsi

2.2.5 Basis Data

Basis data adalah kumpulan data yang saling berkaitan, berhubungan yang disimpan secara bersama-sama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data ini mengandung semua informasi untuk mendukung semua kebutuhan sistem.

Basis data merupakan salah satu komponen yang penting dalam sebuah sistem, karena merupakan basis dalam menyediakan informasi. Basis data menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas,

organisasi data dan perbaharuan data menjadi rumit. Bahasa yang digunakan dalam basis data yaitu [4].

1. DDL (*Data Definition Language*)

   Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek *database* seperti *database,* tabel, *view*. Adapun perintah yang dimiliki adalah:

   a. CREATE: Perintah untuk pembuatan tabel dan *database.*
   b. DROP: Perintah untuk penghapusan tabel dan *database.*
   c. ALTER: Perintah untuk pengubahan struktur tabel yang telah dibuat, seperti menambahkan atribut, mengganti nama atribut ataupun merubah nama tabel.

2. DML (*Data Manipulation Language*)

   Merupakan bahasa manipulasi data yang dilakukan setelah DDL, digunakan untuk memanipulasi data pada objek *database* seperti tabel yang telah dibuat sebelumnya, adapun perintah yang dimiliki DDL adalah:

   a. INSERT: untuk melakukan penginputan atau pemasukan data pada tabel *database.*
   b. UPDATE: Untuk melakukan perubahan/peremajaan terhadap data yang ada pada tabel.
   c. DELETE: Untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset*.

3. DCL (*Data Control Language*)

   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data pada *database,* adapun perintah yang dimiliki DCL adalah:

   a. GRANT: Untuk memberikan hak akses atau izin oleh *administrator server* kepada pengguna.
   b. REVOKE: Untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada *user* oleh *administrator*.

Pengolahan basis data meliputi proses ketiga bahasa yang digunakan diatur dengan menggunakan perangkat sistem manajemen basis data (*Database Management Sistem* / DBMS).

2.2.6 Perangkat Lunak

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*). Pelanggan (*customer*) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang dengan sukarela mengeluarkan untuk memesan atau membeli perangkat lunak. *User* atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya [5].

Secara umum perangkat lunak ini dapat diklasifikasikan menjadi 2, yaitu:

1. Perangkat Lunak Sistem Operasi

   Merupakan perangkat lunak yang berfungsi untuk mengkonfigurasikan komputer agar dapat menerima berbagai perintah dasar yang diberikan sebagai masukan. Perangkat lunak sistem juga merupakan sekumpulan program yang ditulis untuk melayani program-program yang lain. Contoh: *MS-DOS, LINUX, UNIX, FREE BSD, OS/2, SUN OS (JAVA), WINDOWS, MACINTOSH, NOVELL,* dan lain sebagainya.

2. Perangkat Lunak Aplikasi

Merupakan program siap pakai yang digunakan untuk aplikasi dibidang tertentu. Perangkat lunak aplikasi yang membantu *user* sehingga dapat bekerja lebih efektif dan efisien. Aplikasi ini dibagi atas beberapa bidang, antara lain:

1) *Business & Office Application*

Aplikasi perkantoran dipergunakan untuk menyelesaikan pekerjaan kantor seperti mengetik (*Word processing, document management*) membuat tabel kerja (*Worksheet*), membuat Database sederhana (*MS. Access*) ataupun mengolah *image*/citra sederhana. Contoh aplikasi perkantoran yang terpaket komplit: *Mocrosoft Office, Koffice, Star Office, Abiword, Open Office, MS_Office for MAC*, dan lain sebagainya. Aplikasi bisnis lainnya juga meliputi aplikasi pengolahan data keuangan, misal (*Personal Finance Application & Business Acounting Application*). Contoh: Aplikasi perhitungan neraca, rugi laba perusahaan, Aplikasi perhitungan pajak personal & perusahaan, Aplikasi manajemen proyek, Aplikasi persediaan stok barang dan penjualan, dan lain sebagainya.

2) *Database Application*

Aplikasi yang digunakan dalam pengolahan data baik yang berukuran kecil maupun besar bisa digunakan secara *stand-alone* (tunggal) maupun dalam sistem berbasis jaringan lokal *Client server* maupun *webbase* (intranet maupun internet). Contoh: *MS SQL, FoxBase, Oracle, MySQL, PostgresSQL, SQL Server*, dan lain sebagainya.

3) *Graphics Design Application*

Aplikasi yang digunakan untuk membuat sketsa dua dimensi untuk desain ruangan, perabotan, mesin-mesin model industri, juga model tubuh manusia, hewan, dan lain sebagainya. Contoh: *AutoCad, Pro Design, CorelDraw, Adobe Ilistrator,* dan lain sebagainya.

Ada juga yang memasukan dalam golongan ini aplikasi *image, audio, video manipulator*, yaitu aplikasi yang digunakan untuk mengolah citra, suara hingga gambar 3D untuk pembuatan animasi, pengolahan *image, photo, audio, video editing*, dan lain sebagainya. Contoh: *Adobe (Photoshop, After Effect, Premire, & Audition), Corel (Draw, RAVE, Photo Paint), Ulead Media studio, GIMP (In Linux), Cool Edit Pro, Sound Force, Macromedia (Flash, Director), 3D Studio Max,* dan lain sebagainya.

4) *Antivirus & Utility Tools*

Software ini digunakan untuk operasi minor (*Background*) yang bekerja di belakang sistem, berguna untuk meningkatkan kinerja sistem operasi atau aplikasi dan juga performa *hardware*. Terdapat 2 jenis utilitas di dalam sistem operasi:

a) *Utilitas Internal,* sudah termasuk dalam paket yang di instalasi dalam sistem operasi seperti *scandisk* di *Windows* atau perintah *fsck* (*file system check*) di OS Linux.
b) *Utilitas eksternal,* yang harus diinstalasi kemudian pada OS, seperti *System Optimizer, PC Tools, Data/File Recovery, PC Maintenance, Internet Utilities, Norton System Work, Registry Clean & Repair, Hardware Repair*.

Sementara untuk proteksi virus dan antivirus, contohnya adalah antivirus produk *Mc Affee, Grisoft, Norton, PandaSoft,* dan lain sebagainya.

5) *Development Tools*

Merupakan sistem bahasa pemrograman komputer yang dilengkapi dengan *compiler* tertentu sebagai sumber media penyusunan program aplikasi. Contoh: *QuikBasic, Clipper, Visual Basic, Turbo Pascal, Visual C++, GCC, Turbo Assembler, Phyton, KDEvelopt, Gdekstop,* dan lain sebagainya.

Termasuk dalam jenis aplikasi ini adalah *web development tools*, yaitu aplikasi yang digunakan untuk membuat/mendesain tampilan website (*homepage*) di internet, seperti *FrontPage*, *Macromedi Dreamweaver, PHPedit, WebPage Editor*, dan lain sebagainya.

6) *Communication Application*

Aplikasi yang digunakan untuk membangun komunikasi antara satu komputer dengan komputer lain dalam jaringan komputer local maupun global. Contoh: *E-mail & Fax Application, Remote Access, Facebook chat, YM, GTalk, MSN, Terminal Emulation, Instant Messaging, Voice Recognition, Web Browser, Internet Telephony,* dan lain sebagainya.

7) Jenis Aplikasi Lain
   a) *Multimedia Player,* seperti *WinAmp, XMMS, Power DVD, JetAudio, Windows Media Player*, dan lain sebagainya.
   b) *Hacker Tools,* seperti *Find Password, Keygen/Crack Software License, Scanning Port,* dan lain sebagainya.
   c) *Network & Security Application,* seperti *Firewall, Intruder Detections, Network Monitoring*, dan lain sebagainya.
   d) *PC Games Apllication*, dan masih banyak lagi.

2.2.7 Unified Modeling Language (UML)

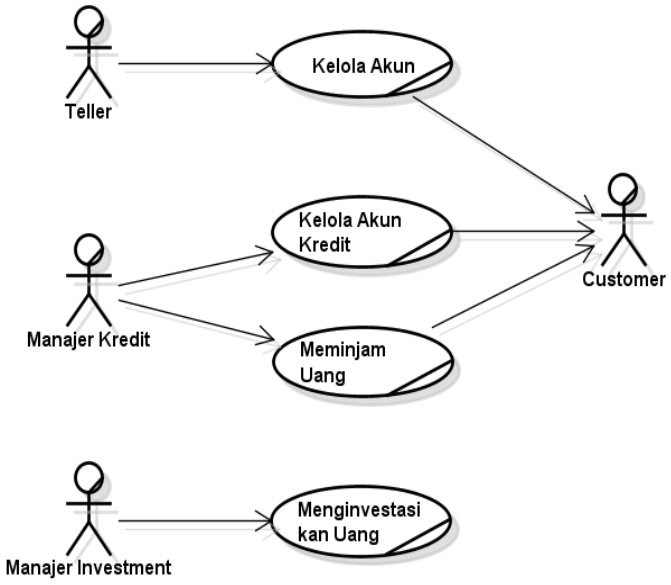
Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar untuk visualisasi, pemodelan, perancangan dan mendokumentasikan artifak suatu sistem perangkat lunak berbasis objek yang terdiri dari kumpulan diagram, dikembangkan untuk membantu para pengembang sistem. Dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh dari Rational Software pada 1990-an. UML telah diadopsi oleh Object Management Group (OMG) pada tahun 2007 dan telah dikelola organisasi ini sejak saat itu. Pada tahun 2000 UML diterima oleh International Organization

for Standardization (ISO) sebagai standar untuk pemodelan sistem perangkat lunak intensif.

Hingga saat ini, UML telah berkembang hingga versi 2.5. UML memiliki banyak sekali diagram, terutama versi terbaru UML yaitu UML 2.5. Berikut akan dijelaskan macam-macam diagram yang paling sering digunakan dalam pembangunan aplikasi berbasis objek [6].

a. *Business Use Case* Diagram

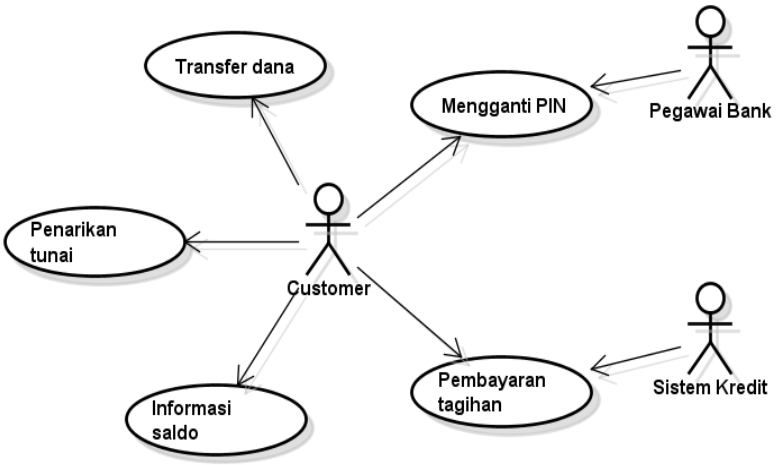
Business use case digunakan untuk mewakili fungsi yang disediakan oleh organisasi secara keseluruhan. *Business use case* juga digunakan secara luas selama kegiatan pemodelan bisnis untuk mengatur konteks sistem dan untuk membentuk sebuah fondasi untuk menciptakan use case.



Gambar 2. 2 Diagram Business Use Case untuk Institusi Finansial [6]

b. *Use Case* Diagram

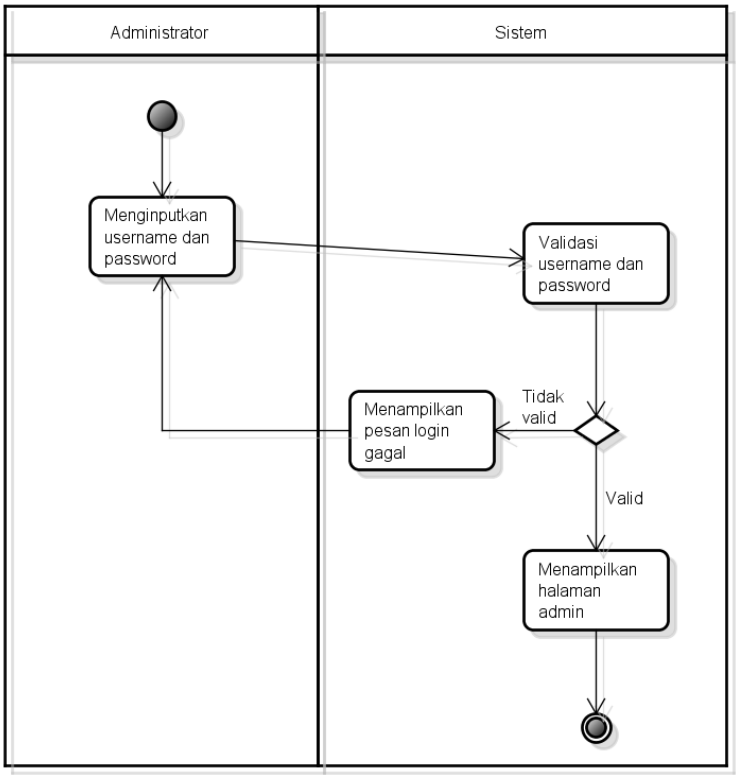
Use case digunakan untuk memodelkan proses bisnis berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk use case dan actor. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.



Gambar 2. 3 Diagram Use Case untuk Sistem ATM [6]

c. *Activity* Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran 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.

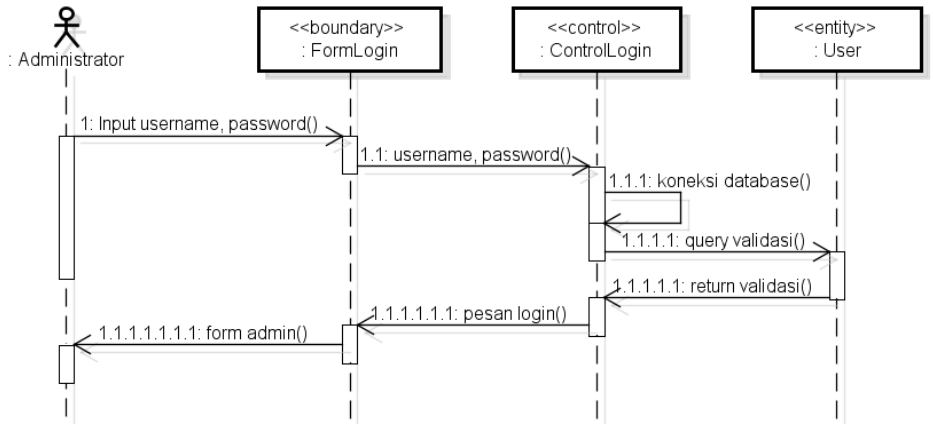


Gambar 2. 4 Diagram Aktifitas untuk Login Administrator [6]

d. *Sequence* Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*, diantaranya:

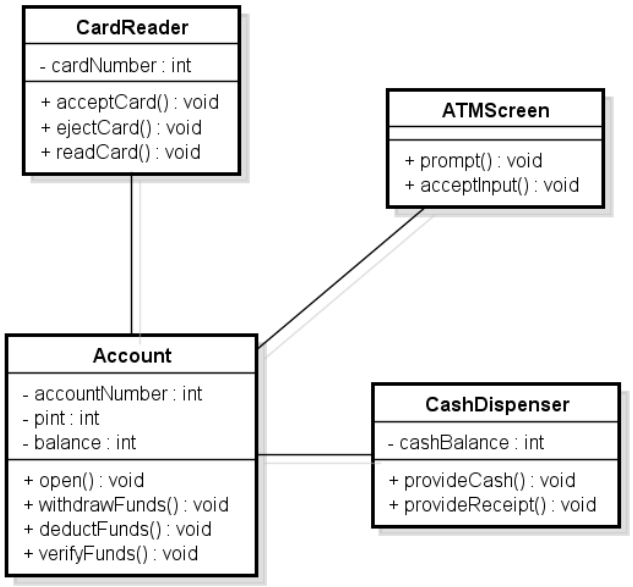
1) Interaksi yang terjadi antar *class*.
2) Operasi apa saja yang terlibat.
3) Urutan antar operasi.
4) Informasi yang diperlukan oleh masing-masing operasi.



Gambar 2. 5 Diagram Sequence untuk Proses Login [6]

e. *Class* Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Selain itu, *class* diagram menunjukan juga atribut dan *method* yang ada pada setiap *class*.



Gambar 2. 6 Class Diagram untuk Use Case Penarikan Uang Sistem ATM [4]

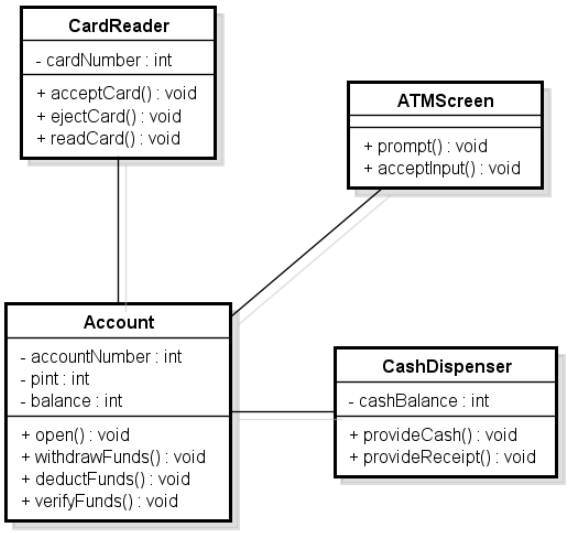
2.2.8 Object Oriented Programming (OOP)

Pemograman terstruktur adalah arus utama pada hari-hari sebelumnya dalam rekayasa perangkat lunak. Programmer mulai mengembangkan blok standar kode untuk melakukan operasi seperti pencetakan, dan kemudian menyalin dan menyisipkan kode tersebut ke setiap aplikasi yang mereka tulis [9]. Sementara hal ini mengurangi waktu pengembangan untuk aplikasi baru, cukup sulit jika perubahan yang diperlukan dalam blok kode, karena pengembang harus membuat perubahan di mana-mana. Dalam pemrograman terstruktur terdapat beberapa hal yang mana pemrograman berorientasi obyek dirancang untuk memecahkan hal tersebut.

Paradigma berorientasi objek adalah cara yang berbeda dalam memandang aplikasi. Pendekatan berorientasi objek, membagi aplikasi ke dalam banyak potongan kecil, atau benda, yang cukup independen satu sama lain. Kemudian aplikasi dibangun dengan menggabungkan semua benda atau potongan kecil ini bersama-sama. Beberapa prinsip orientasi objek diantaranya enkapsulasi, pewarisan, dan polimorfisme.

a. Enkapsulasi (*Encapsulation*)

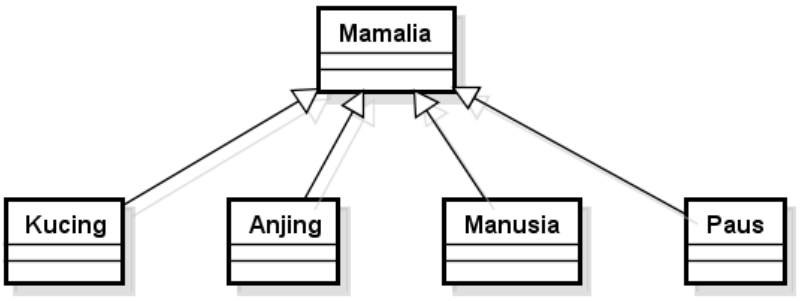
   Sistem berorientasi objek menggabungkan sepotong informasi dengan perilaku tertentu yang bertindak atas informasi tersebut. Lalu dirangkum menjadi sebuah objek. Hal ini disebut dengan enkapsulasi. Cara lain untuk melihat enkapsulasi adalah dengan membagi aplikasi menjadi bagian-bagian kecil dari fungsi terkait. Sebagai contoh, informasi yang berkaitan dengan rekening bank, seperti nomor rekening, saldo, nama pelanggan, alamat, jenis account, tingkat bunga, dan tanggal pembukaan. Lalu perilaku untuk rekening bank: membuka, menutup, deposit, withdraw, jenis perubahan, perubahan pelanggan, dan perubahan alamat. Setelah itu, informasi dan perilaku ini dirangkum menjadi sebuah objek akun. Akibatnya, setiap perubahan ke sistem perbankan mengenai rekening hanya dapat diimplementasikan dalam objek akun. Enkapsulasi bekerja seperti *one-stop shop* untuk semua informasi account dan perilaku.



Gambar 2. 7 Enkapsulasi Model Penarikan Uang di ATM [4]

b. Pewarisan (*Inheritance*)

Pewarisan adalah prinsip dasar berorientasi objek yang kedua. Dalam sistem berorientasi objek, pewarisan adalah mekanisme yang memungkinkan untuk membuat objek baru berdasarkan objek lama: Objek anak mewarisi sifat-sifat dari objek induk. Salah satu manfaat utama dari pewarisan adalah kemudahan dalam pemeliharaan. Ketika suatu perubahan yang mempengaruhi objek anak, hanya perlu mengubah objek orangtua maka objek anak secara otomatis akan mewarisi perubahannya. Contohnya jika mamalia tiba-tiba menjadi berdarah dingin, hanya objek mamalia akan perlu diubah. Kucing, anjing, manusia, ikan paus, dan benda-benda lainnya secara otomatis objek anak akan mewarisi karakteristik baru, yaitu karakteristik berdarah dingin dari mamalia .



Gambar 2. 8 Model Pewarisan Alami [9]

c. Polimorfisme (*Polymorphism*)

Prinsip ketiga orientasi objek adalah polimorfisme. Dari arti katanya, polimorfisme merupakan terjadinya berbagai bentuk, tahapan, atau jenis. Polimorfisme berarti memiliki banyak bentuk atau implementasi dari fungsi tertentu. Dalam hal sistem berorientasi objek, ini berarti bahwa perancang sistem dapat memliliki banyak implementasi dai fungsi tertentu seperti pada contoh kode untuk menggambar bangun ruang berikut.

```
Function Shape.drawMe()
{
SWITCH Shape.Type
Case "Circle"
    Shape.drawCircle();
Case "Rectangle"
    Shape.drawRectangle();
Case "Line"
    Shape.drawLine();
END SWITCH
}
```

Dengan polimorfisme, kode untuk menggambar hanya akan memanggil fungsi drawMe() seperti contoh kode berikut.

```
Function draw()
{
    Shape.drawMe();
}
```

Setiap bentuk (lingkaran, persegi panjang, dan garis) akan memiliki fungsi drawMe( ) untuk menggambar bentuk tertentu. Salah satu manfaat dari polimorfisme sama seperti prinsip-prinsip lain dari orientasi objek, yaitu kemudahan pemeliharaan.

2.2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License*

(GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Berbeda dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Perintah-perintah yang ada di SQL antara lain [4] :

a. Menambah data (*Insert*)

Perintah SQL yang digunakan untuk menambah data ke *database* adalah perintah *INSERT*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
INSERT [INTO] nama_tabel [(daftar field/kolom)]
VALUES (daftar_data).
```

b. Mengakses data (*Select*)

Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*. Bentuk umum perintah ini adalah sebagai berikut:

```
SELECT daftar_select FROM daftar_tabel
[WHERE kondisi_pencarian]
[GROUP BY daftar_group_by]
[HAVING kondisi_pencarian]
[ORDER BY daftar_order[ASC|DESC ]]
```

c. Mengedit/Mengubah data (*Update*)

Perintah SQL yang digunakan untuk mengubah data adalah perintah *UPDATE SET*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
UPDATE nama_table SET field1=databaru1
[,data2=databaru2]
[WHERE kondisi_update].
```

d. Menghapus data (*Delete*)

Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
DELETE FROM tabel_hapus WHERE kondisi_hapus.
```

MySQL memiliki beberapa kelebihan diantaranya [6]:

1. MySQL sebagai *Data Management System* (DBMS) dan *Relation Databese Management System* (RDBMS).
2. MySQL adalah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
3. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini dapat menghubungkan ke media *internet* sehingga dapat diakses jarak jauh.
4. MySQL merupakan sebuah *database client*. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada *server*, jadi MySQL dapat juga berperan sebagai *Client*.
5. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
6. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
7. MySQL didukung oleh driver ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
8. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.
9. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan berisi belasan *field*. Serta MySQL mendukung *field* yang dijadikan sebagai kunci *primer* dan kunci unik atau (*Unique*).
10. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-*update*-an tabel.

2.2.10 Sejarah Bahasa Pemrograman C#

Saat awal sistem operasi windows diperkenalkan oleh Microsoft Corp beberapa belas tahun lalu untuk menggantikan sistem operasi DOS (Disk Operating System), para programer pada umumnya menggunakan bahsa pemograman C untuk membuat program aplikasi yang akan dapat berjalan dengan baik diatas sistem operasi yang bersangkutan. Sebab utamanya adalah karena API (*Application Programing Interface*) *windows* yang banyak dimanfaatkan oleh program-program aplikasi itu juga ditulis menggunakan bahasa C sehingga kompatibilitasnya terjamin. Sementara hal itu dapat dilakukan dengan baik oleh para programer berbakat,untuk sebagian besar programer merasa kesulitan untuk penulisan bahasa pemograman C, sebab saat menulis programer harus melakukan alokasi memori secara manual. Selain itu, kompleksitas bahasa pemograman C masih ditambah dengan adanya pointer-pointer aritmatika yang kompleks serta struktur program yang relatif sukar dibaca. Bahasa C sesungguhnya merupakan bahasa pemograman terstruktur, dimana kita tidak bisa memanfaatkan kelebihan bahasa pemograman yang memiliki paradigma Pemograman Berorientasi Objek (OOP – *Object Oriented Programing*) yang berkaitan dengan penggunaan ulang komponen (*component reusable*), seperti penyembunyian atau pembungkusan informasi (*encapsulation*), pewaris (*intheritance*), polimorfisme, dan sebagainya.

Selanjutnya, di masa-masa berikutnya, para programer bahasa C beralih ke bahasa pemograman C++ untuk melakukan pembangunan program di atas sistem operasi windows. Dalam hal ini bahasa pemograman C++ dapat dianggap sebagai perluasan dari bahasa pemograman C yang beradaptasi dengan konsep pemograman berorientasi objek. Bahasa pemograman C++ ini sama rumitnya dibanding bahasa pemograman C, meski memiliki keunggulan dalam hal adaptasinya dengan paradigma pemrograman berorientasi objek. Meski sangat rumi, beberapa *framework* C++ masih hadir saat ini, misalnya MFC (*Microsoft Foundation Classes*) yang menyediakan kelas-kelas yang dapat digunkan untuk mengembangkan aplikasi *Win32* API (*Application Programming Interface*).

Dalam hal di atas, *Microsoft Corp* cukup tanggap dalam menghadapi keluhan para programer akan kesulitan bahasa pemograman C/C++, yaitu dengan mengembangkan bahasa pemograman Visual Basic yang sangat populer dalam hal kemampuannya dan kemudahannya dalam mengembangkan antarmuka pengguna yang kompleks. Selain itu, *Microsoft Corp* juga menyediakan pustaka-pustaka kelas yang lengkap dalam bentuk COM (*Compounent Object Module*), serta logika akses basis data sangat mudah, bagi para pemrogram Visual Basic. Di atas MFC (*Microsoft Foundation Classes*), Visual Basic menyembunyika kompleksitas Win32 API dari pandangan pemrogram. Meski demikian, versi terakhir bahasa pemograman Visual Basic (VB 6) tidak secara sempurna beradaptasi dengan konsep pemograman berorientasi objek (OOP) sehingga mengurangi fleksibilitas dan pengunaan ulang komponennya.

Sementara itu, di sudut dunia pemograman yang lain, Sun Microsystem Inc mengembangkan bahasa pemograman Java yang memiliki sintaks cukup mirip dengan bahasa C/C++. Dalam hal ini, meskipun bahasa pemograman Java merupakan bahasa pemograman yang elegan dalam adaptasinya dengan pemograman berorientasi objek, integrasi antar program yang dikembangkan menggunakan bahasa pemograman java cukup sukar dilakukan karena sifatnya yang *open source,* yang mana setiap vendor di seluruh dunia memiliki varian produk yang kadang tidak kompatibel satu sama lain. Sebagai contoh, untuk mengimplementasikan konsep ORM (*Object Relational Mapping*), berbagai vendor di seluruh dunia mengembangkan kelas Hibernate, TopLink, iBaTIS, yang saling tidak kompatibel satu sam lain. Ini lain dengan produk *Microsoft Corp* juga mengembangkan objek – objek COM (*Component Object Module*) sedemikian rupa sehingga bisa diakses oleh pemrogram C/C++, Delphi, dan sebagainya, sehingga integrasi antar aplikasi dapat berlangsung dengan mulus. Meski demikian, COM (*Component Object Module*) juga memiliki kelemahan. Dalam hal ini, integraqsi antar aplikasi sepenuhnya berlangsung mulus pada aplikasi Internet yang di dalamnya terkandung berbagai teknologi yang beragam. Dari kenyatan ini, muncullah *framework* .NET yang berusaha sedemikian rupa

sehingga integrasi antar aplikasi ini dapat berjalan dengan lebih mulus dan lebih sempurna lagi [6].

2.2.10.1 Bahasa Pemrograman C# Dan Platform .NET

Bahasa pemrograman C# (baca: C-Sharp) dirancang oleh Microsoft Corp. sebagai bahasa pemrograman yang sangat berdaya guna, aman (*secure*), serta mudah digunakan. Sebagai bagian dari platform .NET, bahasa pemrograman C# dirancang sedemikian rupa untuk bekerja dengan sangat baik di atas framework .NET yang mampu digunakan untuk menulis perangkat lunak handal demi layanan yang cepat. Bahasa pemrograman C# juga dapat digunakan untuk mengembangkan aplikasi sarana bergerak (*mobile application*), aplikasi berbasis Web (*Web-based application*), serta aplikasi berskala besar (*enterprise*).



Gambar 2. 9 Para Penulis Spesifikasi Bahasa C#

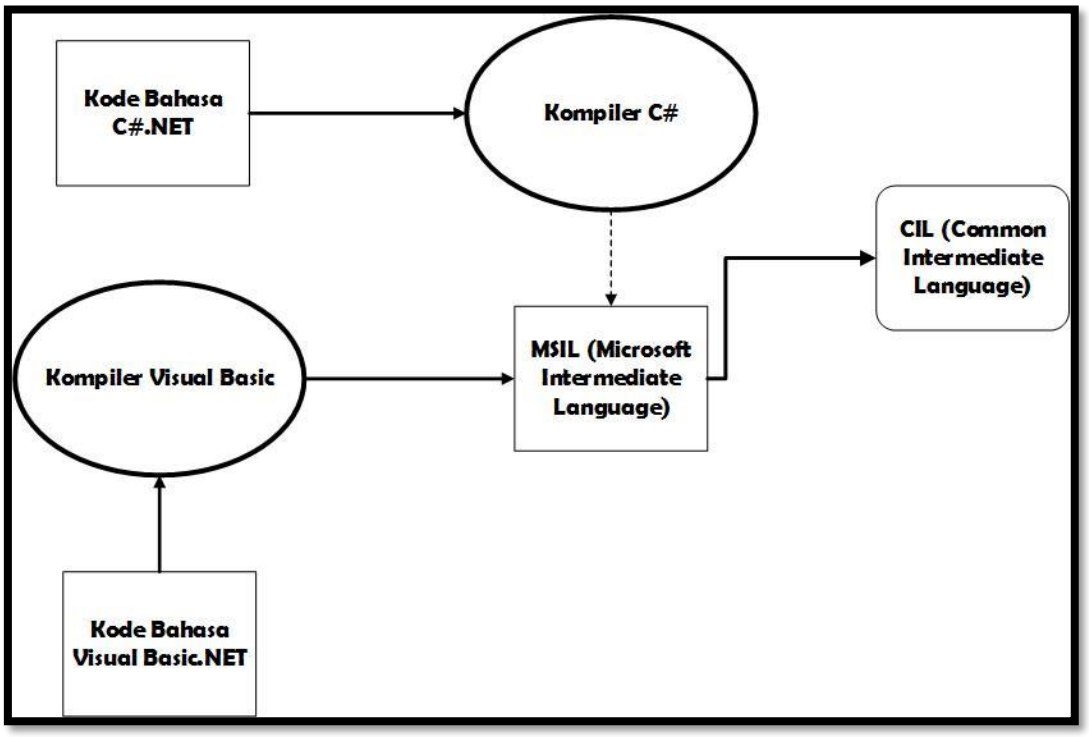
Selanjutnya mungkin ada pertanyaan mengapa kita membahas dan menggunakan bahasa pemrograman C# dalam buku ini ? penulis memilih bahasa pemrograman C# yang diimplementasikan oleh Microsoft Corp. dalam bentuk IDE (*Integrated Development Environment*) Visual C# sebagai basis penulisan buku ini karena bahasa pemograman C#, yang sesungguhnya merupakan salah satu dari bahasa keluarga C (C,C++, *Objective-C*, Java, dan sebagainya) terkenal dengan kecanggihannya, namun secara pemrograman mirip dengan bahasa pemrograman Visual Basic, yang terkenal dengan kemudahannya. Bahasa

pemrograman C# merupakan gabungan dari kecanggihan bahasa keluarga Cdengan kemudahan pemrograman bahasa pemrograman Visual Basic.

Dalam hal di atas, spesifikasi awal bahasa pemrograman C# pada awalnya ditulis oleh Anders Hejsberg dan Scott Wiltamuth dari Microsoft Corp. dengan menggunakan bahsa pemrograman C# kita dapat mengembangkan logika aplikasi yang rmit seperti yang dapat dilakukan menggunakan bahasa keluarga C, namun juga dapat membuat antar muka pengguna yang 'ramah' terhadapa pengguna aplikasi (*user friendly*), serta dapat mengakses sistem basis data relasional (RDBMS – *Relational Database Management System*) dengan cara yang relatif mudah. Hal inilah yang membuat bahasa C# suatu saat akan menjadi salah satu bahasa pemrograman yang paling banyak digunakan di seluruh dunia.

Selanjutnya kita akan mengenal lebih jauh hal-hal yang berkaitan dengan bahasa pemrograman C#. Secara umum bahasa pemrograman C# memiliki karakteristik seperti yang tertulis di bawah ini:

1. Tidak ada alokasi memori secara manual menggunakan pointer
2. Manajemen memori otomatis menggunakan salah satu fiturnya yang dinamakan *garbage collection.*
3. Mendukung kontruksi kelas (*class*), antarmuka (*interface*), struktur, dan enumerasi seperti bahasa-bahasa pemograman berorientasi objek lainnya.
4. Mendukung pemrograman berbasis atribut (*ttribute-based programing*).
5. Mendukung LINQ (*Language Integrated Query*) yang memungkinkan aplikasi yang ditulis menggunakan bahasa pemrograman C# mampu berinteraksi dan bekerja bersama berbagai jenis format data di mana hal ini sangat penting saat kita membuat aplikasi bahasa C#yang mengakses sistem basis data relasional (RDBMS).
6. Mendukung tipe data dan kelas generik
7. Mendukung operator delegasi



Gambar 2. 10 Konversi Kode C# ke Microsoft Intermediate Language [7]

Saat ini kita melakukan kompilasi terhadap program yang kita tulis menggunakan bahasa pemograman C#, kompiler bahasa pemograman C# secara otomatis akan menerjemahkan kode-kode C# yang kita tulis menjadi suatu bentuk kode antara, yan disebut sebagai MSIL (*Microsoft Intermediate Language*), yang sama seperti kita menulis kode menggunakan Visual Basic .NET, Visual C++, dan sebagainya, sebelum kemudian menerjemahkan kembali menjadi berkas-berkas berekstensi .exe atau .dll di atas sistem operasi. Inilah alasan mengapa kita bisa menulis program dalam C#, Visual Basic, sedemikian rupa sehingga mereka masing-msing dapat saling memanfaatkan. *Multifle assembly* dalam hal ini merupakan proses kompilasi beberapa berkas .NET yang masing-masing dinamakan module dimana didalamnya juga berisi berkas *assembly manifest* yang memuat intruksi-intruksi CIL (*Common Intermediate Language*) dan metadata-metadata terkait.

Lalu komponen apa yang memungkinkan kode-kode bahasa pemograman C# dikompilasi agar kompatibilitas antar bahasa keluar .NET tetap terjaga ?

Komponen yang dimaksud adalah *Just In Time Compiler* (JITers) yang bertindak sedemikian rupa sehingga saat MSIL *(Microsoft Intermediate Language*) akan dieksekusi, CLR (*Common Language Runtime*) akan memanggil JITers yang akan menerjemahkan kode-kode bahasa keluarga .NET menjadi kode-kode yang spesifik terhadap komputer tertentu. JITers sesungguhnya peduli terhadap komputer-komputer tertentu yang akan digunakan untuk menjalankan program serta sistem operasi yang digunakannya. Dalam hal ini Microsoft Corp berani menjamin bahwa kode-kode bahasa dari platform .NET akan di eksekusi dengan cara yang lebih cepat dibanding dengan kode-kode yang ditulis menggunakan bahasa C/C++ tradisional.

Tidak seperti kode bahasa C/C++ tradisional, kode yang ditulis menggunakan bahasa pemrograman C# tidak secara langsung diterjemahkan ke bahsa mesin/komputer. Kompiler bahasa pemograman C# mengonversi kode yang ditulis menggunakan bahasa C# menjadi berkas MSIL, yang dinamakan *asembly*. Berkas MSIL dapat dibuat menggunakan semua bahasa pemrograman yang dikembangkan di atas CLR (*Common Language Runtime*), misalnya Visual C++, Visual Basic, serta Visual J#. Sekali lagi inilah alasa mengapa kita bisa menulis program dalam C#, Visual Basic, dan sebagainya, sedemikian rupa sehingga mereka masing-masing dapat memanfaatkan [7].

2.2.11 JSON

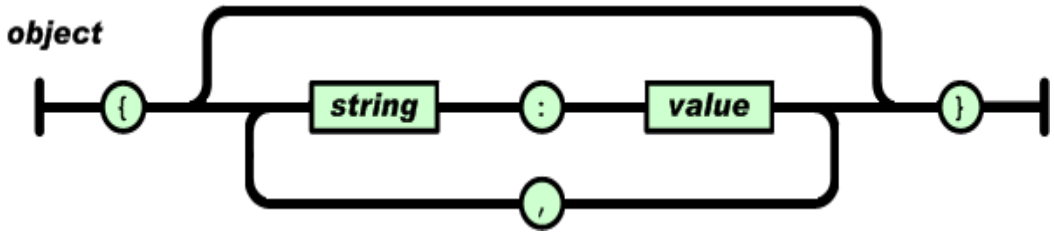
JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

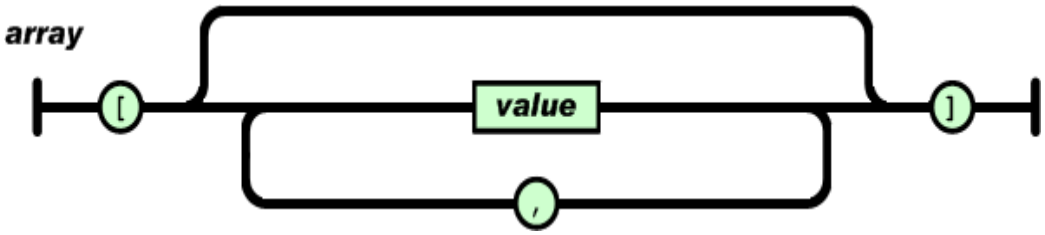
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut [8]:

1. Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



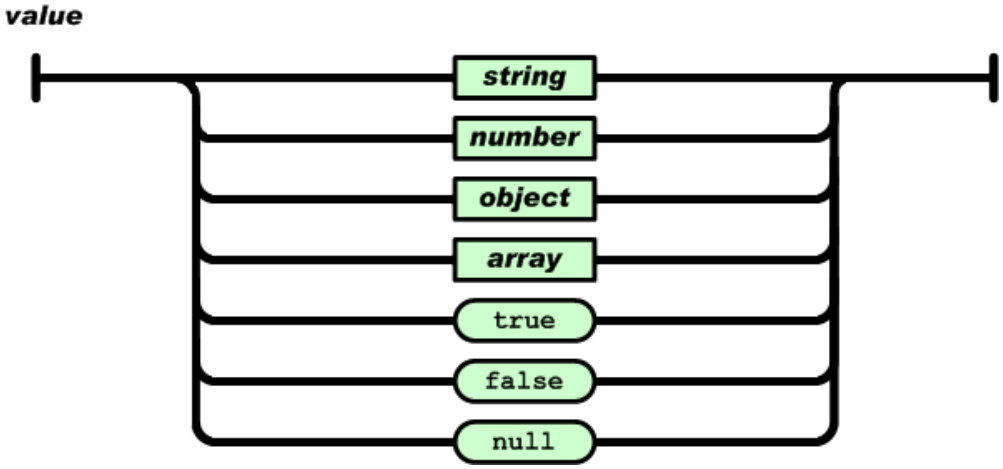
Gambar 2. 11 Penulisan Objek Pada C# [8]

2. Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



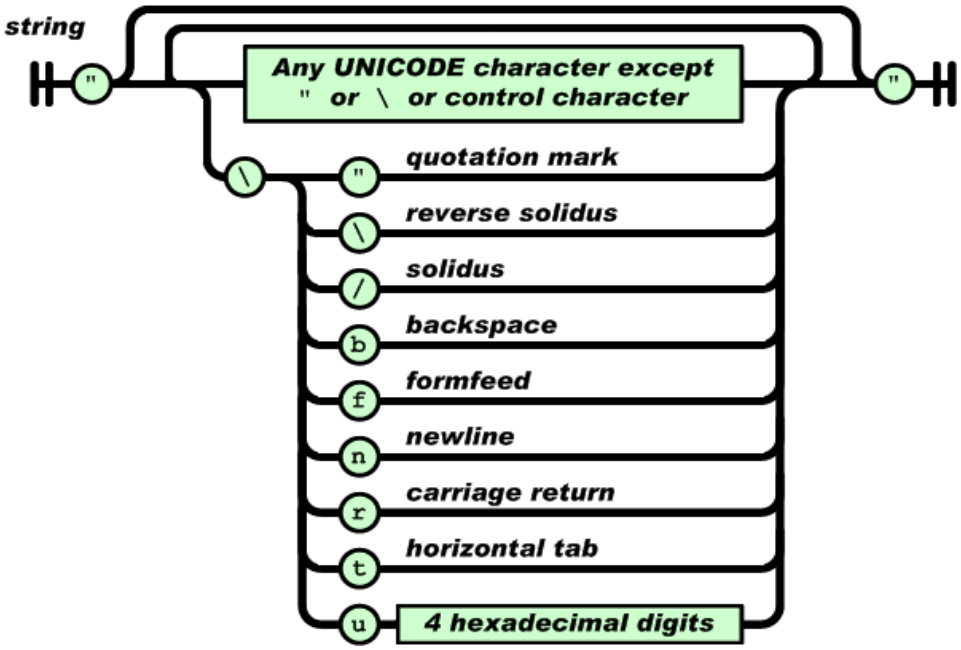
Gambar 2. 12 Penulisan Array Pada C# [8]

3. Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau *angka*, atau true atau false atau null, atau sebuah *objek* atau sebuah *larik*. Struktur-struktur tersebut dapat disusun bertingkat.



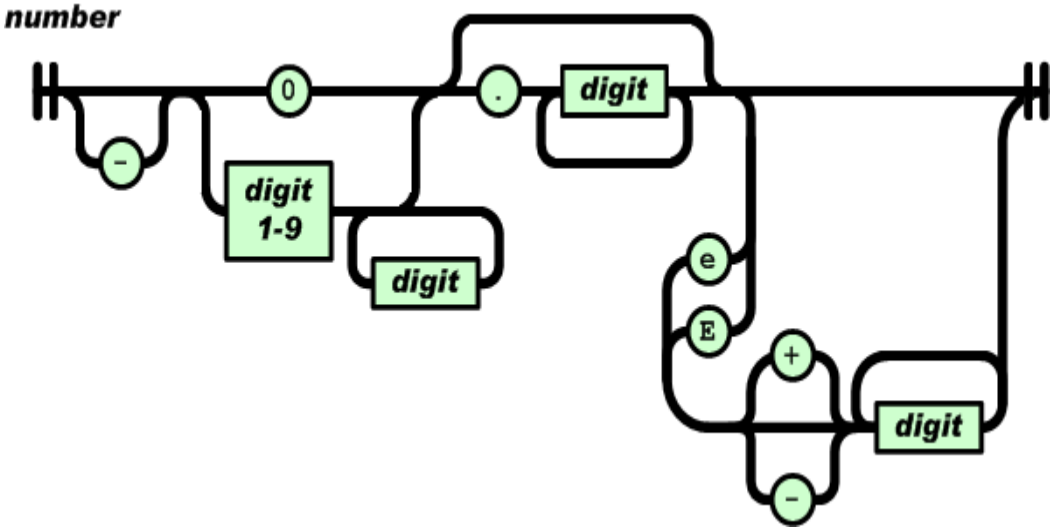
Gambar 2. 13 Nilai Pada C# [8]

4. String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar 2. 14 Format String Pada C# [8]

5. Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2. 15 Format Angka Pada C# [8]

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil *encoding* yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan.

2.2.12 Radio Frequency Identification (RFID)

RFID (*Radio Frequency Identification*) adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut label RFID atau transponder (tag) 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[13].

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 *tag* RFID

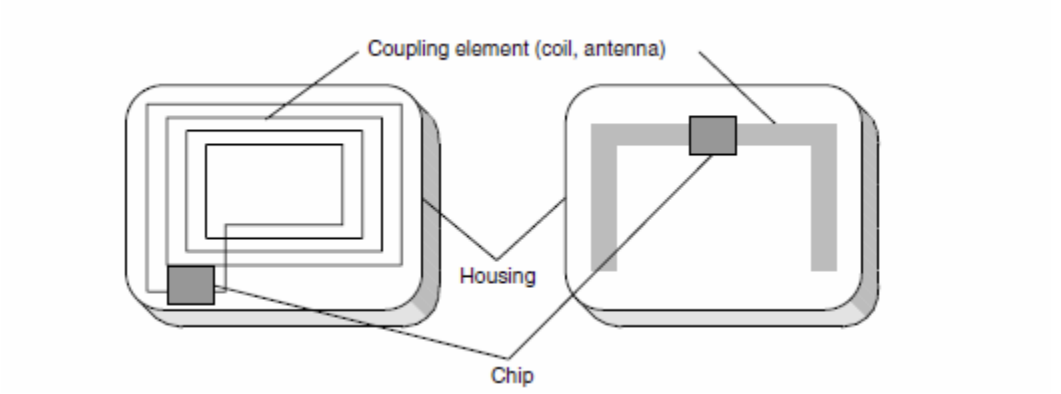
2.2.12.1 Komponen Komponen RFID

RFID terdiri atas 3 komponen utama, yaitu:

1. RFID tag, 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 tag .
3. Host computer, terdiri dari atas Basis data yang menyimpan semua data yang terkandung dalam tag. Sistem computer yang mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara tag dan reader

2.2.12.2 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 2.16 menunjukkan layout dasar dari sebuah RFID *tag*:



Gambar 2. 16 Layout Dasar RFID tag [9]

2.2.12.3 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, tag 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.

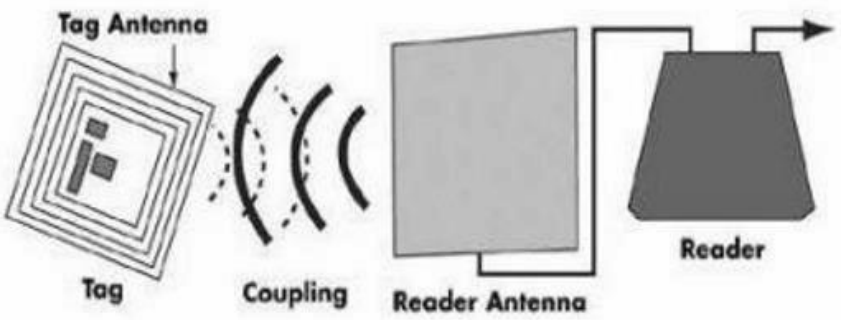
Dalam fungsinya, RFID reader dituntut untuk dapat melakukan dua tugas, yaitu berkomunikasi melalui gelombang radio dan membaca data yang dibawa tag kemudian diteruskan ke aplikasi.



Gambar 2. 17 RFID reader

2.2.12.4 Konfigurasi Umum

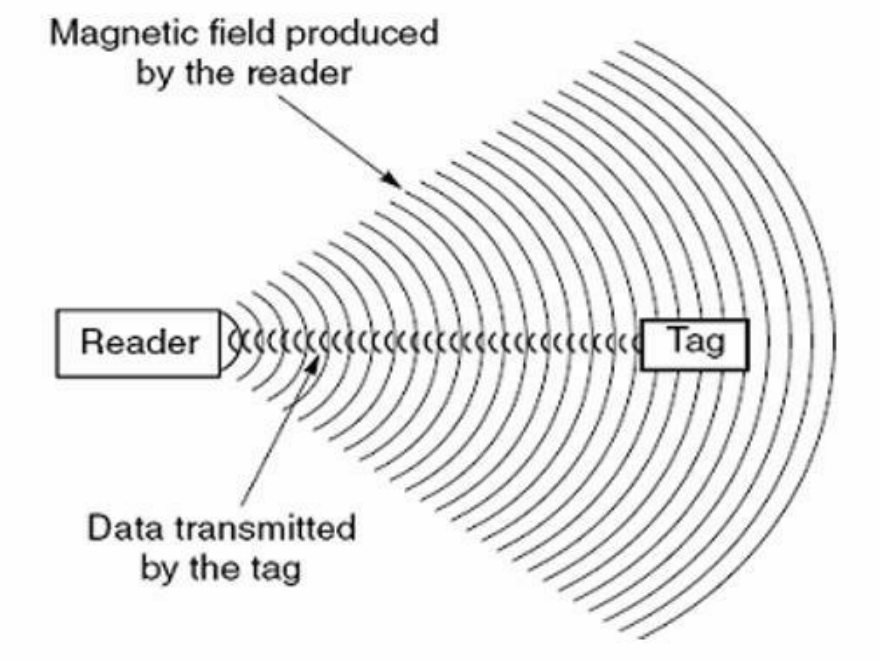
Konfigurasi umum RFID tersebut minimal memerlukan sebuah tag (yang berfungsi sebagai transponder), sebuah reader (yang berfungsi sebagai interrogator), dan sebuah antenna (yang berfungsi sebagai coupling device). Reader biasanya terhubung dengan sebuah host computer atau perangkat lainnya yang memiliki kecerdasan untuk memproses lebih lanjut tag data dan memutuskan untuk mengambil suatu tindakan. Salah satu elemen penting pada RFID adalah data transfer. Data transfer terjadi ketika terjadi hubungan antara sebuah tag dengan sebuah reader, yang dikenal dengan coupling, melalui antenna baik yang terpasang pada tag tersebut maupun pada reader seperti yang diilustrasikan pada gambar berikut ini.



Gambar 2. 18 Hubungan antara tag, reader dan antena

2.2.12.5 Metode Coupling

Coupling pada kebanyakan sistem-sistem RFID menggunakan metode magnetic (inductive) atau electromagnetic (backscatter). Metode yang digunakan tersebut bergantung pada harga, ukuran, kecepatan, dan jangkauan pembacaan serta keakuratan. Pada umumnya komunikasi antara sebuah tag dengan sebuah reader terjadi melalui sebuah physical principle yang dikenal sebagai sebuah backscatter modulation. Pada proses ini, sebuah reader mengirimkan sinyal kepada sebuah tag, dan tag akan menanggapinya dengan memantulkan sebagian dari energi ini kembali ke reader. Hal ini dapat diilustrasikan dari gambar dibawah ini [10]:



Gambar 2. 19 Data transfer pada RFID tag dan RFID reader

2.2.13 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 development 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.