BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Perusahaan

Mizan - dalam bahasa Arab berarti seimbang- didirikan pada tahun 1983 oleh tiga mahasiswa beserta dua senior mereka. Pada awalnya mereka bertujuan membangun aras baru karya-karya ke-Islaman di Indonesia. Dimulai hanya menerbitkan buku-buku terjemahan penulis terkemuka dari luar negeri, secara bertahap karakter Mizan menemukan bentuknya melalui karya-karya ke-Islaman yang mewakili berbagai sudut pandang, secara serius dan modern.

Untuk menjawab kebutuhan perkembangan bisnis yang berkembang pesat, Mizan membentuk Mizan Publika sebagai perusahaan induk (*holding company*) pada tahun 1999, dan meluncurkan program rekayasa ulang (*re-engineering*) di tahun 2001. Sejak saat itu, Mizan terbagi dalam unit-unit yang otonom dan dinamis sehingga dapat beradaptasi terhadap cepatnya perubahan lingkungan bisnis. Program rekayasa-ulang melahirkan beragam unit bisnis strategis yang terbagi dalam lini penerbitan, distribusi, penjualan, galeri buku online, percetakan, dan *learning center* untuk pelatihan-pelatihan.

Dari hanya menerbitkan tiga buku sebulan di tahun pertamanya, Kelompok Mizan telah mampu memproduksi lebih dari 600 judul buku per tahunnya dan sukses menghasilkan karya-karya best-seller. Mizan saat ini dikenal sebagai salah satu perusahaan penerbitan terbesar di Indonesia.

Saat ini Mizan tidak hanya memenuhi kebutuhan pasar domestik, memperluas jangkauan bisnisnya ke luar negeri melalui unit bisnis strategis khusus yang diberi nama Mizan International dengan tiga bidang kegiatan. Pertama, mendistribusikan buku-buku Mizan berbahasa Indonesia dan Melayu kepada masyarakat Indonesia dan Melayu yang tinggal di negeri jiran; kedua, menerjemahkan buku-buku berbahasa Indonesia ke dalam bahasa asing untuk pasar internasional; dan ketiga menangani urusan hak cipta (*copyright*) melalui lembaga bentukannya, yaitu *Mizan Literary Agency (MLA*).

Mizan kini telah mulai merintis pembukaan jaringan pusat kegiatan buku di Indonesia, termasuk toko buku dan pusat kegiatan Komunitas Mizan sebagai wadah penulis, penerbit, sastrawan, dan pihak-pihak lain dalam komunitas baca (*reading society)* di kalangan masyarakat Indonesia. Selanjutnya jaringan pusat perbukuan ini akan dikembangkan di Jakarta dan kota besar lainnya.

Di masa mendatang Mizan juga berencana untuk go public melalui Bursa Efek Indonesia setelah memperbesar asetnya melalui *private placement*.

2.1.1 Logo Perusahaan



Gambar 2.1 Logo Perusahaan

2.1.2 Profil Perusahaan

Profil perusahaan dimana praktian melaksanakan penelitian, sebagai berikut :

Nama Perusahaan	: PT.MIZAN PUSTAKA
Alamat	: Jln.Cinambo No.135 Cisaranten Wetan, Ujung Berung Bandung 40249
Web site	: www.mizan.com
Badan Hukum	: *Perseroan Terbatas (PT)*

2.1.3 Visi Misi Perusahaan

Mizan telah menunjukkan komitmen yang meningkat dalam ikut serta menciptakan masyarakat modern, terbuka, dan, pada saat yang sama, religius. Komitmen ini diwujudkan melalui pengembangan perusahaan yang berjalan berlandaskan pengetahuan, yang mampu merespon tantangan dan permintaan seiring perubahan lingkungan yang cepat dan mengglobal akibat kemajuan teknologi dan informasi.

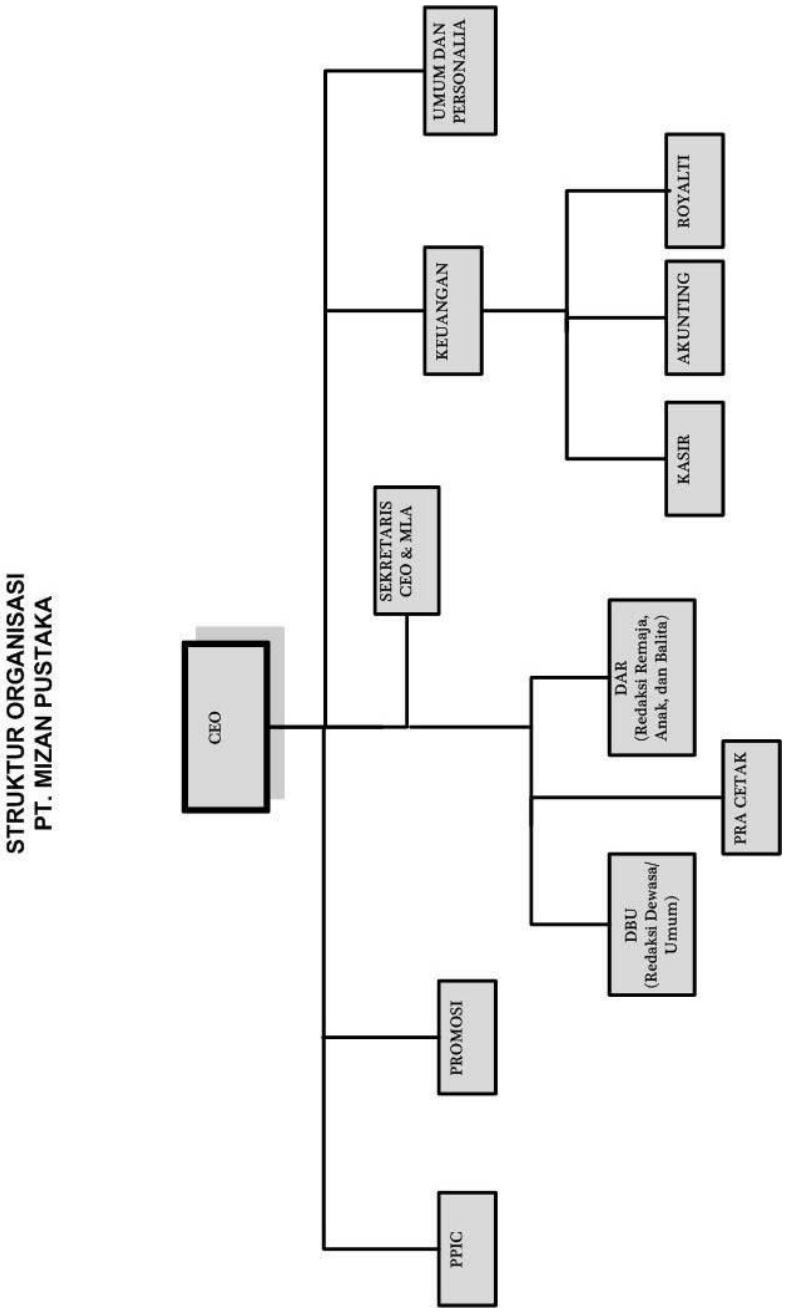
2.1.4 Bidang Usaha

Pada awalnya PT. Mizan Pustaka bertujuan membangun aras baru karya-karya ke-Islaman di Indonesia. Dimulai hanya menerbitkan buku-buku terjemahan penulis terkemuka dari luar negeri, secara bertahap karakter Mizan menemukan bentuknya melalui karya-karya ke-Islaman yang mewakili berbagai sudut pandang, secara serius dan modern. Dan hingga kini PT.Mizan Pustaka merintis pembukaan jaringan pusat kegiatan buku di Indonesia, termasuk toko buku dan pusat kegiatan Komunitas Mizan sebagai wadah penulis, penerbit, sastrawan, dan pihak-pihak lain dalam komunitas baca (*reading society*) di kalangan masyarakat

Indonesia. Selanjutnya jaringan pusat perbukuan ini akan dikembangkan di Jakarta dan kota besar lainnya.

2.1.5 Struktur Organisasi Perusahaan

Struktur organisasi dalam PT. Mizan Pustaka terbagi dalam bebarapa golongan dengan kekuasaan tertinggi dipegang oleh seorang CEO (*Chief Executive Officer*). CEO dalam organisasi ini memiliki peranan penting untuk memimpin dan bertanggung jawab dalam perusahaan.



Gambar 2.2 Struktur Organisasi PT.Mizan Pustaka

Adapun tugas dan fungsi masing-masing bagian adalah sebagai berikut :

1. CEO *(Chief Executive Officer)*
   a) Memimpin seluruh dewan
   b) Memimpin rapat umum, dalam hal: untuk memastikan pelaksanaan tata-tertib; keadilan dan kesempatan bagi semua untuk berkontribusi secara tepat; menyesuaikan alokasi waktu per item masalah; menentukan urutan agenda; mengarahkan diskusi ke arah konsensus; menjelaskan dan menyimpulkan tindakan dan kebijakan
   c) Bertindak sebagai perwakilan organisasi dalam hubungannya dengan dunia luar
   d) Memfasilitasi bisnis diluar perusahaan selain membimbing para pegawai dan eksekutif lainnya kearah tujuan utama . Seorang CEO harus memiliki suatu keseimbangan internal dan eksternal dalam membangun suatu perusahaan yang berkelanjutan.
   e) Memainkan bagian terkemuka dalam menentukan komposisi dari board dan sub-komite, sehingga tercapainya keselarasan dan efektivitas
   f) Mengambil keputusan sebagaimana didelegasikan oleh BOD atau pada situasi tertentu yang dianggap perlu, yang diputuskan, dalam meeting-meeting BOD.
2. Umum dan Personalia
   1) Penarikan pegawai baru, yang meliputi :
      a) memelihara arsip lamaran pekerjaan
      b) memelihara hubungan dengan lembaga-lembaga penghasil calon pegawai.

c) mengadakan seleksi calon pegawai.

2) Processing data pegawai baru, meliputi:

a) penyiapan surat keputusan pengangkatan, dan penempatan dalam golongan gaji bagi pegawai baru

b) penyiapan laporan penarikan tenaga kerja ke Departemen Tenaga Kerja.

3) Pengembangan pegawai

4) Pemberhentian pegawai

3. Keuangan

Bertanggung jawab atas pengambilan keputusan penting mengenai investasi (investment) dan pendanaan (financing).

Fungsi dan tugas keuangan :

a) Merencanakan dan menganalisa pembelanjaan perusahaan

b) Mengatur struktur aktiva (struktur kekayaan perusahaan)

c) Mengatur struktur financial

d) Melakukan pengawasan atas biaya

e) Menetapkan kebijaksanaan harga

f) Meramalkan laba yang akan datang

g) Mengukur atau menjajaki biaya modal kerja

h) Menyediakan Laporan keuangan (Neraca, Laporan Rugi/Laba, dan Laporan Perubahan Modal).

4. Promosi dan Marketing

a) Manajer pemasaran bertanggung-jawab terhadap manajemen bagian pemasaran

b) Manajer pemasaran bertanggung-jawab terhadap perolehan hasil penjualan dan penggunaan dana promosi

c) Manajer pemasaran sebagai koordinator produk dan manajer penjualan

d) Manajer pemasaran membina bagian pemasaran dan membimbing seluruh karyawan dibagian pemasaran

e) Manajer pemasaran membuat laporan pemasaran kepada direksi

5. Manajer Publishing

a) Bertanggung jawab terhadap buku-buku yang akan dan telah diterbitkan

b) Mengatur jumlah buku yang akan diterbitkan

c) Membuat laporan tentang buku-buku yang diterbitkan kepada direksi

d) Menyeleksi naskah-naskah yang masuk, untuk diterbitkan

6. Sekretaris CEO dan MLA

a) Menyusun dan membuat jadwal kegiatan pimpinan.

b) Membuat surat-surat dan menyimpan arsip/surat masuk dan keluar

c) Menerima telepon yang berhubungan dengan naskah yang masuk

d) Mengonsep surat perjanjian kerjasama dengan relasi atau instansi luar.

e) Menghadiri rapat-rapat dinas, sebagai pendamping pimpinan selama mengadakan pertemuan bisnis.

f) Mengadakan pemeriksaan peralatan kantor, mana yang perlu diperbaiki dan mana yang tidak perlu diperbaiki atau penambahan alat-alat dan sarana kantor.

7. DBU (Redaksi Dewasa Umum)

   Sebagai pemimpin yang memimpin team buku spiritual dan bertanggung jawab atas buku-buku dalam bidang agama atau spritual.

8. DAR (Redaksi Anak, Balita dan Remaja)

   Sebagai pemimpin yang memimpin team buku anak dan bertanggung jawab atas buku anak-anak.

9. PPIC *(Production Planning Inventory Control)*

   Sebagai menerima order dari bagian Penjualan (Sales/marketing) lalu memastikan order ini selesai dan dikirim ke customer pada waktu yang sudah disepakati.

10. PRA CETAK

    Sebagai pemimpin yang memimpin team Layout dan bertanggung jawab atas desain-desain yang akan diterbitkan atau dicetak.

11. Editor

    Editor bertugas melakukan pengeditan atau penyuntingan naskah sehingga menjadi lebih menarik dan disukai oleh pembaca.

12. Graphic Designer

    Sebagai Graphic designer, ada beberapa jenis tugas seperti ilustrator (menggambar sketsa), tracing (penebalan gambar dengan tinta), colouring

(pewarnaan, dengan manual atau digital). Selain ahli menggambar, dibutuhkan pula web desainer untuk mendukung promosi melalui internet. Seorang desainer harus dapat menguasai software *seperti after effects, 3dmax, corel draw, photoshop,* dan sebagainya.

13. Staff Pemasaran dan Marketing
    a) Membantu manajer dalam mengurus agenda pertemuan penulis dengan pembaca
    b) Mengelola web yang berhubungan dengan pemasaran buku
    c) Mengelola *social media* untuk menginformasikan buku-buku terbaru atau menjawab pertanyaan seputar buku oleh pembaca
    d) Berhubungan dengan customer. Pengaduan ataupun saran dan kritik.

2.2 Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

"Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau bentuk penyelesaian suatu sasaran tertentu [2]."

Pendekatan sistem merupakan jaringan kerja dari prosedur yang lebih menekankan urutan operasi di dalam sistem. Prosedur disini menurut Richard F. Neuschel adalah

"Suatu urutan operasi *klerikal* (tulis-menulis), yang biasanya melibatkan beberapa orang dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi [3]."

Lebih lanjut dalam bukunya, Jerry FitzGerrald, Ardra F. FitzGerald dan Warren D. Stallings, Jr, mendefinisikan prosedur sebagai berikut :

"Prosedur adalah urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan dan bagaimana (*who*) mengerjakannya."

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefiniskan sistem sebagai berikut :

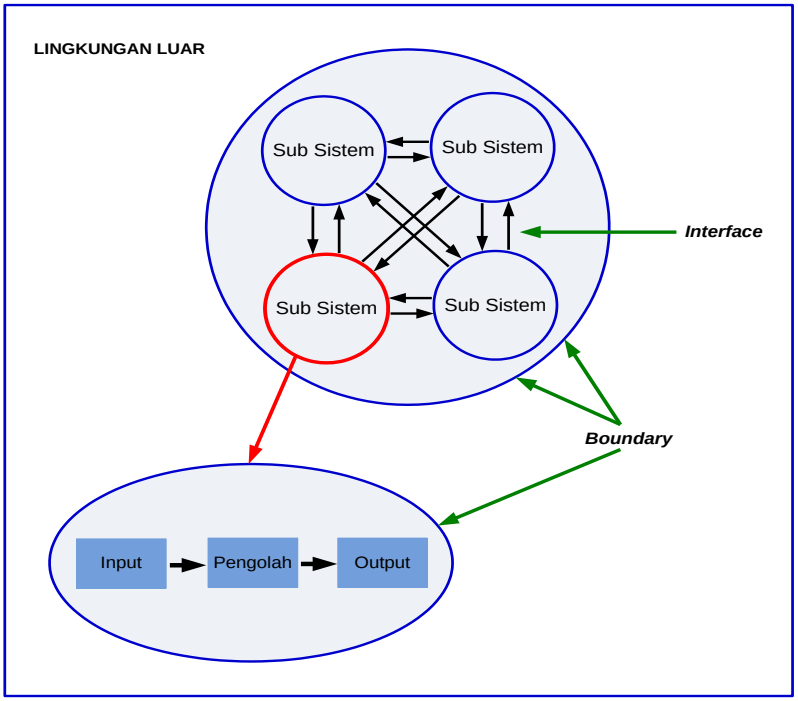
"sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu."

Kedua kelompok definisi ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya. Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang luas. Definisi inilah yang banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau bagian-bagian sistem. Suatu sistem mempunyai maksud tertentu. Ada yang menyebutkan maksud dari suatu sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*). Perbedaan tujuan dan sasaran terletak pada ruang lingkup yang dilibatkan, jika yang

dilibatkan adalah sistem bisnis maka istilah “tujuan” lebih cocok untuk dijadikan maksud dari sistem itu sendiri, tetapi jika sistem akuntansi sebagai contoh lingkup yang dilibatkannya, maka sasaran lebih cocok untuk dijadikan maksudnya.

2.2.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar system (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan tujuan (*goal*) atau sasaran (*objectives*).



Gambar 2.3 Karakteristik Sistem

2.2.2 Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem yang atau elemen-elemen sistem dapat berupa suatu subsistem

atau bagian-bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, Selalu mempunyai sifat-sifat dari sistem untuk menjalanka suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut *supra system.*

2.2.3 Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan sistem dipandang sebagai satu kesatuan. Batas uatu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

2.2.4 Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendailakn, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

2.2.5 Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penguhubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*)

untuk subsistem yang lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

2.2.6 Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* merupakan energi yang dimasukan supaya sistem tersebut beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran (*output*). Sebagai contoh didalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

2.2.7 Keluaran Sistem

Keluaran (*output*) adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan merupakan keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

2.2.8 Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukkan berupa bahan baku menjadi keluaran berupa barang jadi.

2.2.9 Sasaran dan Tujuan Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objectives*). Tujuan dari sistem sangat mempengaruhi dan menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.10 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai **sistem abstrak (*abstract system*)** dan **sistem fisik (*physic system*).** Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem computer, sistem akuntansi, sistem produksi dan lain sebagainya.
2. Sistem diklasifikasikan sebagai **sistem alamiah (*natural system*)** dan **sistem buatan manusia (*human made system*).** Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebutnya dengan *man-machine system*. Sistem informasi merupakan contoh *man-machine system,* karena menyangkut penggunaan komputer yang berinterkasi dengan manusia.

3. Sistem diklasifikasikan sebagai **sistem tertentu (*deterministic system*)** dan **sistem tak tentu (*probabilistic system*).** Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankannya. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasikan sebagai **sistem tertutup (*closed system*)** dan **sistem terbuka (*open system*).** Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar sistem. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar sistem, maka suatu sistem harus mempunyai pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.

2.3 Basis Data

Basis data (*database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi. Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.[4]

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang

lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak *administrator* dan *programmer* menggunakan istilah basis data untuk kedua arti tersebut.

DBMS (*Database Management System*) adalah *software* yang menangani semua akses ke basis data. Secara konsep apa yang terjadi adalah sebagai berikut :

1. User melakukan pengaksesan basis data untuk informasi yang diperlukannya menggunakan suatu bahasa manipulasi data, biasanya disebut SQL.
2. DBMS menerima *request* dari user dan menganalisa request tersebut
3. DBMS memeriksa skema *eksternal user*, pemetaan eksternal/konseptual, skema konseptual, pemetaan konseptual/internal, dan struktur penyimpanan.
4. DBMS mengeksekusi operasi-operasi yang diperlukan untuk memenuhi permintaan *user*.

2.3.1 Bahasa Basis Data

DBMS merupakan perantara bagi pemakai dengan basis data dalam disk. Cara berkomunikasi/berinteraksi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat *DBMS*. Bahasa tersebut dapat disebut sebagai bahasa basis data yang terdiri atas sejumlah perintah yang diformulasikan oleh *user* dan diproses oleh *DBMS* untuk melakukan suatu aksi atau pekerjaan tertentu.

Ada 3 bahasa yang digunakan dalam basis data yaitu :

1. DDL (*Data Definition Language*)

   Merupakan bahasa definisi data yang digunakan untuk membuat dan *memanage* objek *database* seperti *database*, tabel dan *view*.

2. DML (*Data Manipulation Language*)

   Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek *database* seperti tabel.

3. DCL (*Data Control Language*)

   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.

2.3.2 *Object Oriented Programming (OOP)*

OOP diciptakan karena dirasakan masih adanya keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah, semua pemecahan masalah dibagi ke dalam objek. Dalam OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai objek. Proses perancangan atau desain dalam suatu pemrograman merupakan proses yang tidak terpisah dari proses yang mendahului, yaitu analisis dan proses yang mengikutinya. Pembahasan mengenai orientasi objek tidak akan terlepas dari konsep objek seperti *inheritance* atau penurunan, *encapsulation* atau pembungkusan, dan *polymorphism* atau kebanyakrupaan. Konsep-konsep ini merupakan fundamental dalam orientasi objek yang perlu sekali dipahami serta digunakan dengan baik, dan menghindari penggunaannya yang tidak tepat.

Ada beberapa konsep utama OOP antara lain :

1. **Class** : kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of mobil' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari mobil.
2. **Objek** : membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek. Contoh nya : Mobil, Motor, Sepeda
3. **Abstraksi** : kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4. ***Inheritance*** atau biasa disebut dengan istilah pewarisan, dalam konsep ini kita dapat membuat suatu class baru yang bentuknya dapat disamakan dengan class yang sebelumnya. Merupakan bentuk yang lebih spesifik dari Object misalnya Mobil makainheritance dari mobil adalah dapat berupa Mobil Truk Tronton, Mobil Truk FUSO, Mobil Sedan, Mobil Pick up.

5. ***Encapsulation*** biasa disebut dengan istilah membungkus atau mengemas sebuah informasi yang akan menghasilkan sebuah objek. Tujuannya agar informasi yang dihasilkan tidak menyebar keluar dari aturan yang ada.
6. ***Polymorphism*** menghasilkan banyak bentuk. Konsep ini mengembangkan konsep pada pewarisan dimana kita dapat membuat dan menghasilkan sebuah class baru yang dapat diperluas kembali menjadi banyak bentuk.[5]

2.3.3 *Unified Modeling Language (UML)*

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO(*Object-Oriented*). [6] UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software. *Unified Modelling Language* (UML) juga merupakan sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendeskripsikan OOP (*Object Oriented Programming*) dengan beberapa diagram, diantaranya:

- **Diagram struktur:**
    1. Diagram kelas *(Class Diagram)*
    2. Diagram obyek *(Object Diagram)*
    3. Diagram komponen *(Component Diagram)*

4. Diagram deployment *(Deployment Diagram)*

- **Diagram perilaku:**

1. Diagram use-case *(Usecase Diagram)*
2. Diagram urutan/sekuen *(Sequence Diagram)*
3. Diagram kolaborasi *(Collaboration Diagram)*
4. Diagram statechart *(Statechart Diagram)*
5. Diagram aktivitas *(Activity Diagram)*

2.4 Internet

Internet merupakan jaringan global komputer dunia**,** besar dan sangat luas sekali dimana setiap komputer saling terhubung satu sama lainnya dari negara ke negara lainnya di seluruh dunia dan berisi berbagai macam informasi, mulai dari text, gambar, audio, video, dan lainnya yang terhubung melalui protokol TCP/IP. Dalam mengatur integrasi dan komunikasi jaringan komputer ini menggunakan protokol yaitu TCP/IP**.** TCP (Transmission Control Protocol) bertugas untuk memastikan bahwa semua hubungan bekerja dengan benar, sedangkan IP (Internet Protocol) yang mentransmisikan data dari satu komputer ke komputer lain. TPC/IP secara umum berfungsi memilih rute terbaik transmisi data, memilih rute alternatif jika suatu rute tidak dapat di gunakan, mengatur dan mengirimkan paket-paket pengiriman data [7].

2.5 World Wide Web (WWW)

Website atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun

dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink) dokumen web yang ditampilkan dalam *browser* web.

2.5.1 Sejarah Web

Sejarah Web bermula Maret 1989 di European Laboratory for Particle Physics (lebih dikenal dengan nama CERN), di kota Geneva dekat perbatasan Perancis dan Swiss. CERN (*Consei European pour la Receherche Nuclarie*) merupakan suatu organisasi yang didirikan oleh 18 negara di Eropa, peneliti lainnya dari CERN mengusulkan suatu protokol sistem distribusi informasi di Internet yang memungkinkan para anggotanya yang tersebar di seluruh dunia saling membagi informasi dan bahkan untuk menampilkan informasi tersebut dalam bentuk grafik. Pada tahun yang sama CERN dan MIT kemudian mendirikan suatu konsorsium yang dinamakan World Wide Web Consortium (W3C) yang bertugas untuk membangun standar bagi Web. [8]

2.5.2 Aplikasi Web

Aplikasi web pada awalnya dibangun hanya dengan menggunakan bahasa yang disebut dengan HTML dan protokol yang dinamakan *Hypertext Transfer Protocol* (HTTP). Pada perkembangan berikutnya sejumlah *script* dan objek dikembangkan untuk memperluas kemampuan HTML, antara lain *script* PHP dan *Active Server Pages* (ASP), sedangkan contoh berupa objek adalah Applet (Java). Aplikasi web sendiri dapat dibagi menjadi dua, yaitu :

1) Web statis, dibentuk dengan HTML saja. Kekurangan aplikasi ini hanya terletak pada keharusan untuk memelihara program secara terus menerus untuk megikuti perubahan yang terjadi.
2) Web dinamis, yang terdiri dari *server-side scripting* seperti *script* ASP dan PHP dan *client-side scripting* seperti Vbscript ataupun Javascript. Web dinamis mengatasi kekurangan web statis karena aplikasi web yang dinamis memanfaatkan kemampuan *server* web untuk mengakses sistem database.

2.6 Electronic Book (*E-Book)*

E-Book adalah singkatan dari Electronic Book atau buku elektronik. E-book tidak lain adalah sebuah bentuk buku yang dapat dibuka secara elektronis melalui komputer. E-book ini berupa file dengan format bermacam-macam, ada yang berupa pdf (portable document format) yang dapat dibuka dengan program Acrobat Reader atau sejenisnya. Ada juga yang dengan bentuk format html, yang dapat dibuka dengan browsing atau internet eksplorer secara offline. Ada juga yang berbentuk format exe.[9]

Sebuah *e-book*, sebagaimana didefinisikan oleh Oxford Kamus bahasa Inggris, adalah "*versi elektronik dari buku cetak yang dapat dibaca pada komputer pribadi atau perangkat genggam yang dirancang khusus untuk tujuan ini*". *E-book* didedikasikan bagi mereka para pembaca media elektronik atau perangkat e-book baik melalui komputer atau bisa juga melalui ponsel yang dapat digunakan untuk membaca buku elekronik ini.

Dengan hadirnya *e-book* ini para pembaca dimudahkan untuk tidak menyimpan buku-buku favoritnya dalam bentuk fisik (buku konvensional) dan juga memudahkan bagi para penulis dalam menyebarkan tulisan-tulisannya, karena melalui ebook ini seseorang tidak perlu datang ke penerbit hanya sekedar menginginkan tulisannya dapat diterbitkan. Apabila seorang penulis ingin menjual atau mempublikasikan tulisan-nya dengan adanya e-book ini merupakan salah satu jalan pintasnya dan ini berlaku juga bagi para pembaca atau pencari ilmu di internet.

2.7 *Text To Speech (TTS)*

2.7.1 Sejarah *Text To Speech*

Penelitian di bidang pesintesa ucapan mengalami perjalanan yang sangat panjang dan telah dimulai sejak lama. Salah satu catatan literatur awal yang berhubungan dengan sintesa ucapan adalah pernyataan seorang ahli matematika dan *engineer* terkenal yang bernama **Leonhard Euler** pada tahun 1791. Euler menyatakan *It would be a conciderable invention indeed, that of a machine able to mimic speech, with its sounds and articulations. I think it isno imposible.* Menurut beliau suatu hari nanti akan ada penemuan yang pantastis yaitu suatu mesin yang mampu meniru suara dengan bunyi yang serasi, itu bukan suatu hal yang mustahil.

Cara interaksi manusia dengan komputer mengalami perkembangan dari waktu ke waktu. Saat ini senagian besar komputer masih menggunakan *keyboard* sebagai piranti utama untuk memasukan data dan layar monitor sebagai piranti

utama untuk menyajikan hasil komputasi. Berkembangnya sistem operasi berbasis grafik, seperti Microsoft Windows, telah menyebabkan digunakannya piranti *mouse* sebagai pelengkap *keyboard*.[10]

Keinginan untuk membuat cara berinteraksi yang lebih dialami, kecenderungan ukuran fisik komputer yang semakin kecil, serta tingkat mobilitas pengguna yang semakin tinggi menyebabkan perlunya dicari alternatif lain sebagai pengganti atau pelengkap piranti interaksi yang selama ini digunakan. Manusia sendiri banyak menggunakan ucapan untuk berinteraksi antara manusia.

Cara interaksi menggunakan ucapan dianggap lebih mudah, cepat dan lebih nyaman dilakukan. Untuk mewujudkan keinginan tersebut, ada tiga teknologi kunci yang diperlukan, yaitu sebagai berikut:

1. Teknologi *speech recognition* (pengenalan ucapan), diperlukan untuk mengenali setiap ucapan menjadi teks.
2. Teknologi *natural language processing* (NLP), diperlukan, untuk menggantikan ucapan-ucapan yang sudah dikenali oleh bagian pengenal ucapan dan mengkonversikannya ke dalam format yang dapat diolah oleh komputer. Hal ini diperlukan karena bahasa lisan yang diucapkan secara langsung sering tidak mengikuti kaidah-kaidah bahasa formal.
3. Teknologi *speech synthesizer* (pensintesa ucapan), diperlukan, untuk mengucapkan informasi-informasi teks yang dihasilkan oleh komputer. Sistem seperti ini dikenal pula dengan istilah sitem *Text To Speech* (TTS).

Berdasarkan hasil studi literatur dari berbagai sumber bacaan, perkembangan teknologi pensintesa ucapan dapat dibagi menjadi tiga kurun

waktu. Kurun waktu pertama adalah sebelum tahun 1930. Pada masa ini penelitian-penelitian lebih banyak ditujukan untuk memahami karakteristik sinyal ucapan serta pengembangan pensintesa ucapan berbasis mekanik atau elektromekanik. Kurun waktu kedua dimulai sejak tahun 1930-an sampai dengan ditemukannya komputer digital. Masa ini ditandai dengan pengambangan berbagai alat pensintesa ucapan menggunakan teknologi elektronik analog. Kurun waktu ketigadimulai sejak ditemukannya komputer digital hingga sekarang. Pada masa ini, sintesa ucapan dilakukan menggunakan pendekatan-pendekatan pemrosesan sinyal digital.

Teknologi komputer telah memperlihatkan kemajuan yang luar biasa pada pertengahan abad ini. Pada tahun 1945 tidak ada komputer dengan program tersimpan. Saat ini, hanya dengan uang beberapa ribu dolar saja dapat digunakan untuk membeli komputer yang mempunyai kelebihan unjuk kerja, dengan *main memory* yang lebih besar serta penyimpan yang lebih besar dibandingkan dengan komputer yang dibeli pada tahun 1965 dengan harga satu juta dolar. Kemajuan yang sangat pesat ini, diperoleh dari kemajuan teknologi yang digunakan untuk membuat komputer serta dari inovasi dalam perancangan sistem komputer.

Sejak tahun 1985, industri komputer memperlihatkan pola arsitektur baru yang memberikan beberapa kelebihan dan memulai periode dimana unjuk kerjanya semakin meningkat dengan kecepatan yang lebih tinggi. Melalui kemajuan dalam teknologi *compiler* dan ide arsitektur baru, perancang mampu menciptakan serangkaian komputer dengan perbaikan unjuk kerja dengan faktor hampir dua kali lipat.

2.7.2 Konversi Teks ke Suara

Sistem *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu :

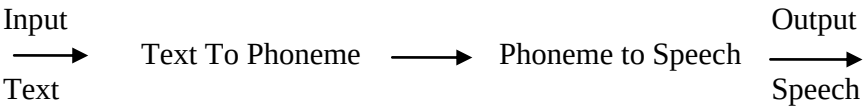
1) Bagian Konverter Teks ke Fonem (*Text to Phoneme*)

Bagian Konverter Teks ke Fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. Bagian ini bersifat sangat *language dependant*. Untuk suatu bahasa baru, bagian ini harus dikembangkansecara lengkap khusus untuk bahasa tersebut.

2) Bagian Konverter Fonem to Ucapan (*Phoneme to Speech*)

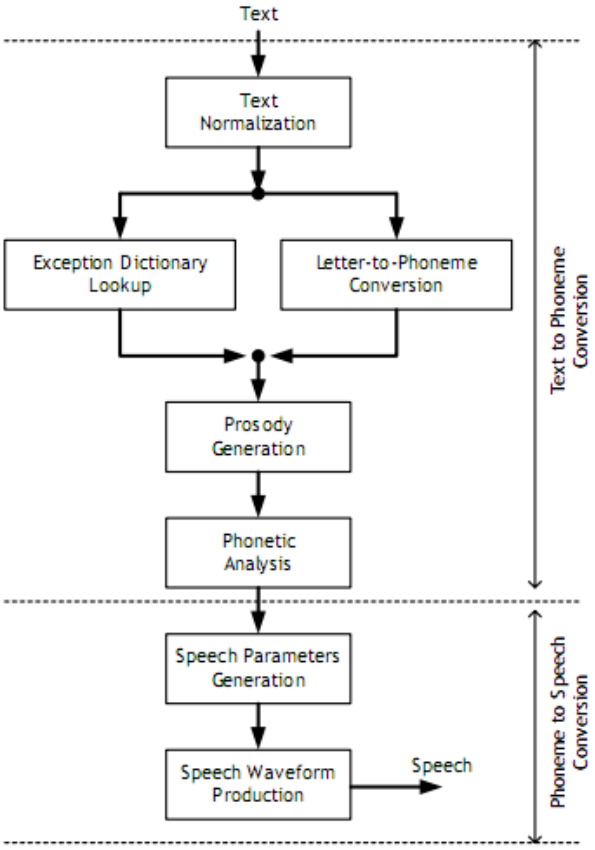
Bagian Konverter Fonem ke Ucapan akan menerima masukan berupa kode-kode fonem serta pitch dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah *formant synthesizer*,serta *diphone concatenation*. Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Pada sistem yang menggunakan teknik *diphone concatenation*, sistem harus didukung oleh suatu *diphone database* yang berisi rekaman segmen-segmen ucapan yang berupa diphone. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa,

oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda.



Gambar 2.4 Skema Text To Speech

Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda [11]. Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan pada Gambar 2.5.



Gambar 2.5 ***Proses Konversi dari Teks ke Ucapan (dimodifikasi dari Pelton, 1992)***

2.8 *Google Text To Speech API*

Google telah memperkenalkan upgrade ke layanan menerjemahkan dengan menambahkan ketentuan berbicara dengan suara ke teks atau sebaliknya yang diterjemahkan. Saat ini memang tidak ada API resmi untuk layanan text-to-speech. Pada dasarnya, Google adalah mengambil *HTTP GET* biasa (REST) berbasis permintaan dan mengembalikan audio dalam format MP3, Permintaan URL dari :

http://translate.google.com/translate_tts?tl=en&q=some+text

Pada API ini hanya menaruh beberapa teks berbasis REST di URL di atas, dan mendapatkan kembali file berbentuk MP3 [11]. Google menggunakan diphone database *espeak speech synthesizer* dan juga Mbrola. Google API ini juga menyediakan *engine* dalam *mobile*, *windows* dan *website* dengan menggunakan berbagai macam bahasa pemograman.

Google Translate kini sudah memiliki lebih dari 60 bahasa selain digunakan dalam menerjemahkan kata atau kalimat, *Google Translate* juga berfungsi untuk mengkonversi tulisan menjadi suara. Dalam penggunaan URL diatas untuk mendapatkan suara dengan bahasa yang berbeda, hanya mengganti **en** menjadi kode bahasa yang diinginkan. Misalnya pada **ar** untuk bahasa Arab atau **ja** untuk bahasa Jepang.

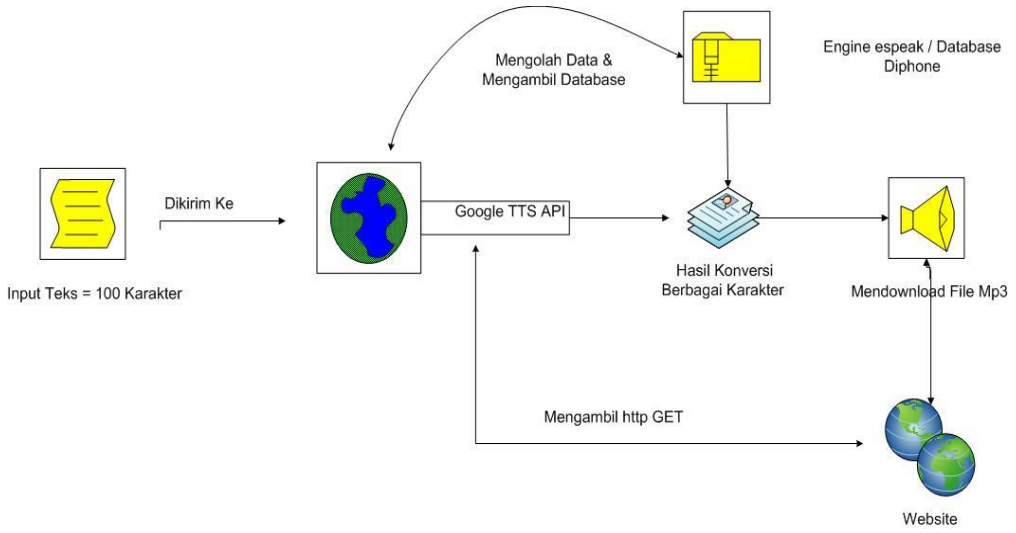
Keterangan dari penggunaan GET url diatas adalah :

q : *query string* untuk mengkonversi tulisan menjadi suara (.mp3)

`tl` : untuk menerjemahkan bahasa, misalnya, *`ar`* untuk bahasa Arab , atau **`en-us`** untuk Bahasa Inggris.

Adapun alur dari Google TTS API adalah :

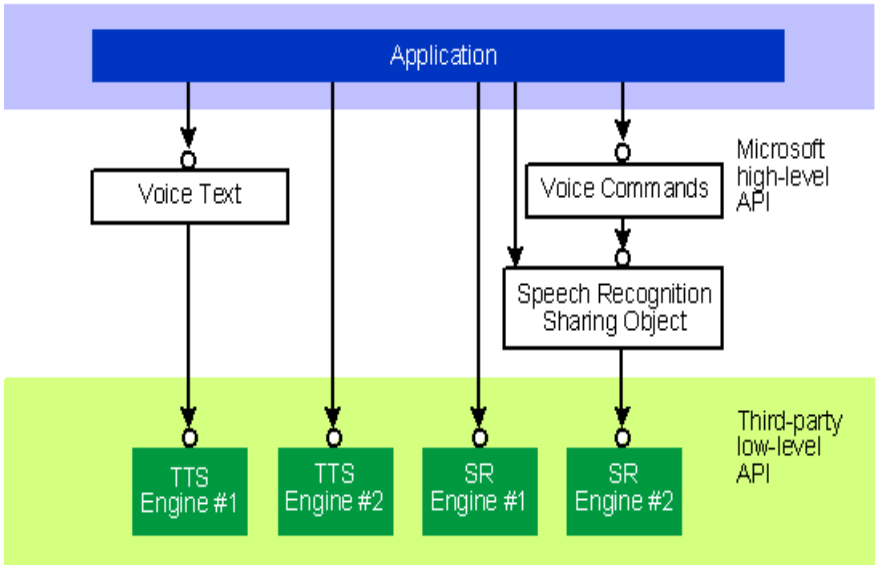
1. *Script* membagi input teks untuk potongan paling banyak 100 karakter. Setiap potongan kemudian dikirim ke TTS Google Api.
2. Lalu Google TTS API membaca atau menerjemahkan tulisan yang dikirimkan melalui *url* tersebut.
3. Setelah itu Google TTS akan mengirimkan output berapa suara berformat mp3.
4. Menginputkan teks di Google TTS ini harus memperhatikan tanda baca dan juga jumlah karaketnya. Titik dan juga koma mempengaruhi artikulasi dan pemotongan pada kalimat.
5. Setelah titik "." waktu jeda akan lebih lama karena titik merupakan akhir dari sebuah kalimat.
6. Tanda baca "!?.," juga menunjukkan akhir potongan pada kalimat yang mengandung unsur seru dan juga tanya, namun waktu jeda tidak terlalu lama.
7. jika kalimat terlalu panjang tanpa tanda baca, maka suara terdengar tidak begitu jelas terutama dalam Bahasa Indonesia.
8. Untuk menghasilkan suara yang baik, lebih baik memodifikasi susunan kalimat yang dimiliki sesuai dengan kebutuhan.



Gambar 2.6 Alur *Text To Speech* Menggunakan Google TTS API

2.9 *Microsoft SAPI*

Speech Application Programming Interface (SAPI) adalah sebuah API yang dikembangkan oleh Microsoft yang digunakan sebagai pengenal suara didalam lingkungan pemrograman aplikasi Windows. Sampai saat ini SAPI dikemas baik berupa SDK (*Sistem Development Kit*) maupun disertakan dalam sistem operasi Windows itu sendiri. Aplikasi yang telah menggunakan SAPI antara lain *Microsoft Office*, dan *Windows Vista*. Secara arsitektur pemrograman SAPI dapat dilihat sebagai sebuah *middleware* yang terletak antara aplikasi dan *speech engine* [12]. Di dalam SAPI versi 1 sampai dengan 5, aplikasi dapat berkomunikasi langsung dengan *speech engine* seperti tampak pada gambar berikut:



Gambar 2.7 Arsitektur Microsoft SAPI

Komponen utama di dalam SAPI 5 adalah sebagai berikut:

a. *Voice Command*, sebuah obyek level tinggi untuk perintah dan kontrol menggunakan pengenalan suara.

b. *Voice Dictation*, sebuah obyek level tinggi untuk continous *dictation speech recognition.*

c. *Voice Talk*, sebuah obyek level tinggi untuk *speech synthesis*.

d. *Voice Telephony*, sebuah obyek untuk menulis aplikasi telepon berbasiskan pengenalan suara.

e. *Direct Speech Recognition*, sebuah obyek sebagai mesin untuk mengontrol pengenalan suara (*direct control of recognition engine*)

f. *Direct Text to Speech*, sebuah obyek sebagai mesin yang mengontrol synthesis.

g. *Audio Object*, untuk membaca dari audio device atau sebuah *file audio*

Option Explicit

SAPI 5.1 terdiri dari 2 antar muka yaitu *application programming interface* (API) dan *device driver interface* (DDI).

2.9.1 Aplication Programming Interface (API)

Pada sistem pengenalan pembicaraan, aplikasi akan menerima even pada saat suara yang diterima telah dikenali oleh *engine*. Komponen SAPI yang akan menghasilkan even ini diimplementasikan oleh antar muka ISpNotifySource. Lebih spesifik, SAPI menggunakan SetNotifySink, yaitu aplikasi akan meneruskan pointer IspNotify- Sink ke ISpNotifySource:: SetNotifySink. ISpNotifySource ::SetNotify. Sink ini akan menerima pemanggilan melalui IspNotify- Sink:: Notify ketika terdapat satu atau lebih even yang menyatakan bahwa aplikasi dapat mengambil data. Biasanya aplikasi tidak mengimplementasikan ISpNotifySink secara langsung tetapi menggunakan CoCreate Instance untuk membuat obyek IspNotify Sink, yang diimplementasikan oleh komponen CLSID_SpNotify. Obyek ini menyediakan antar muka ISpNotifyControl. Tetapi antar muka ISpNotifySource dan ISpNotifySink hanya menyediakan mekanisme untuk notifikasi dan tidak ada even yang ditimbulkan oleh notifikasi tersebut.Ketika aplikasi menerima notifikasi, ada kemungkinan terdapat informasi yang sama pada beberapa even. Dengan memanggil ISpEventSource::GetInfo, maka variable anggota ulCount akan mengembalikan nilai yang berupa struktur SPEVENT SOURCEINFO yang didalamnya terdapat jumlah even yang mempunyai informasi yang sama. Dengan menggunakan IspEvent Source::GetEvents, aplikasi akan mengeluarkan sejumlah mempunyai informasi tentang even tertentu. Ketika terjadi notifikasi pada saat pengenalan

pembicaraan bekerja, maka lParam yang merupakan variabel anggota dari struktur SPEVENT akan menjadi ISpRecoResult yang kemudian digunakan oleh aplikasi untuk dapat menentukan apa yang telah terkenali dan sekaligus menentukan ISpRecoGrammar mana yang harus digunakan. ISpRecognizer, baik *shared* ataupun InProc, dapat mempunyai ISpRecoContext lebih dari satu dan masingmasing ISpRecoContext dapat menerima notifikasi sesuai dengan even yang telah didefinisikan. Sebuah ISpRecoContext dapat mempunyai lebih dari satu ISpRecoGrammars di mana masing-masing ISpRecoGrammar tersebut digunakan untuk mengenali tipe yang berbeda.

2.9.2 Device Driver Interface (DDI)

DDI menyediakan fungsi untuk menerima data suara dari SAPI dan mengembalikan pengenalan frasa pada level SAPI paling dasar. Terdapat dua antar muka yang digunakan oleh DDI yaitu ISpSREngine, yang diimplementasikan oleh *engine* dan ISpSREngineSite yang diimplementasikan oleh SAPI. *Engine* menyediakan layanan ke SAPI melalui antar muka ISpSREngine. Semua fungsi pengenalan terjadi melalui IspSREngine:: RecognizeStream. Ketika SAPI memanggil ISpSREngine::SetSite, maka SAPI memberikan pointer ke antar muka ISpSREngineSite dimana kemudian *engine* dapat berkomunikasi dengan SAPI selama ISpSREngine::RecognizeStream dieksekusi. SAPI membuat sebuah *thread* ke obyek ISpSREngine dan *engine* tidak boleh meninggalkan ISpSREngine::Recognize Stream sampai terjadi kesalahan atau SAPI sudah terindikasi dengan menggunakan ISpSREngineSite::Read, dimana tidak ada lagi data yang dapat diproses dan

engine telah selesai melakukan tugasnya. SAPI memisahkan pembuat *engine* dari kerumitan untuk mengatur peralatan suara secara detail. SAPI menjaga *logical stream* dari *raw audio data* dengan membuat indeks posisi *stream*. Dengan menggunakan indeks posisi *stream*, *engine* dapat melakukan pemanggilan terhadap ISpSREngineSite::Read untuk menerima *buffer* dari *raw audiodata* selama ISpSREngine: :Recognize Stream dieksekusi. Pemanggilan ini akan terjadi sampai semua data yang dibutuhkan tersedia. Jika ISpSREngineSite::Read menghasilkan data yang lebih sedikit dari yang dibutuhkan, yang berarti tidak ada data lagi, maka *engine* akan menghentikan eksekusi ISpSREngine:: RecognizeStream DDI memungkinkan *engine* untuk hanya mempunyai satu buah thread yang dieksekusi antara SAPI dan *engine*. Satusatunya metode yang tidak mengijinkan Isp SREngine untuk masuk dan keluar secara cepat ialah ISpSREngine:: Recognize Stream. [13]

2.10 *Tool*s Yang Digunakan Untuk Membangun Aplikasi

Dalam membangun aplikasi diperlukan beberapa *tool*, yang dijelaskan pada sub bab–sub bab di bawah ini.

2.10.1 .NET

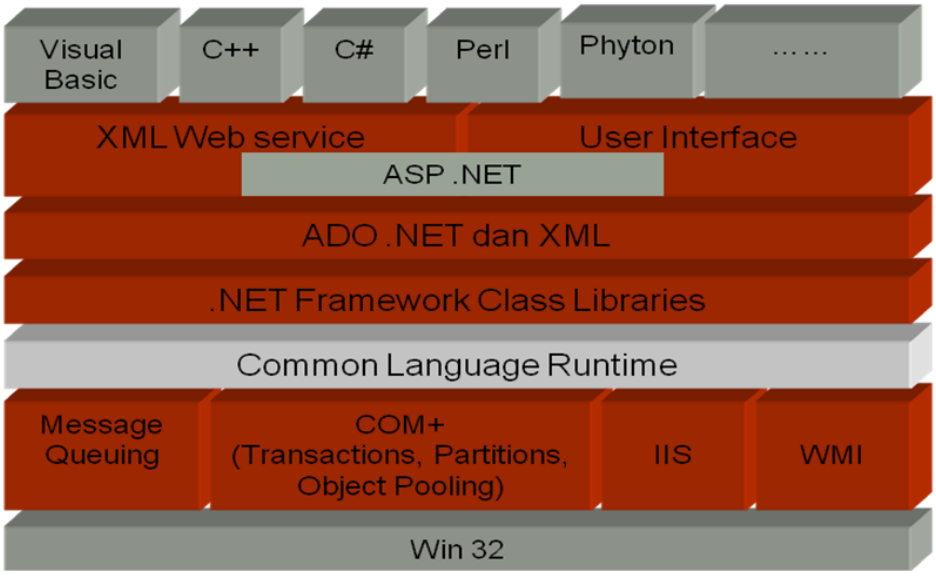
.NET Platform merupakan satu kumpulan teknologi yang dapat mempermudahkan developer untuk membuat aplikasi dalam lingkungan dan *device* yang beragam. Secara teknikal, .NET Platform menyediakan konsep pemrograman dengan *library* dan modul – modul baru yang konsisten, terlepas

dari jenis bahasa pemrograman yang digunakan. .NET Platform menyediakan hal – hal berikut bagi para developer :

1. *Language Independent*, dengan programming model yang konsisten di semua tier aplikasi yang dibangun.
2. *Interoperability* dan kompatibilitas antar aplikasi.
3. Kemudahan migrasi dari teknologi yang ada saat ini.
4. Dukungan penuh terhadap berbagai teknologi standar yang digunakan dalam platform internet, antara lain HTTP, XML, SOAP, dan HTML.

2.10.1.1 Framework .NET

Suatu komponen windows yang terintegrasi yang dibuat dengan tujuan untuk mensupport pengembangan berbagai macam jenis aplikasi serta untuk dapat menjalankan berbagai macam aplikasi generasi mendatang.



Gambar 2.8 Framework .NET

2.10.2 Tujuan Framework .NET

Framework .NET didesign untuk memenuhi beberapa tujuan berikut ini :

1. Untuk menyediakan *environment* kerja didalam mengeksekusi kode yang dapat meminimalisasi proses *software deployment* dan menghindari konflik penggunaan versi software yang dibuat.
2. Untuk menyediakan *environment* kerja yang konsisten bagi bahasa pemrograman yang berorientasi objek (OOP) baik kode objek itu disimpan dan dieksekusi secara lokal, atau dieksekusi secara lokal tapi didistribusikan melalui internet atau dieksekusi secara *remote*.
3. Untuk menyediakan *environment* kerja yang aman dalam hal pengeksekusian kode, termasuk kode yang dibuat oleh pihak ketiga *(third party)*.
4. Membuat para *developer* lebih mudah mengembangkan berbagai macam jenis aplikasi yang lebih bervariasi, seperti aplikasi berbasis windows dan aplikasi berbasis web.
5. Membangun semua komunikasi yang ada didalam standar industri untuk memastikan bahwa semua kode aplikasi yang berbasis Framework .NET dapat berintegrasi dengan berbagai macam kode aplikasi lain.

Sebagai salah satu sarana untuk dapat memenuhi tujuan diatas, maka dibuatlah berbagai macam bahasa pemrograman yang dapat digunakan seperti bahasa C#, VB .NET, J#, Perl .NET , Phyton dan lain – lain.

2.10.2.1 Arsitektur Framework .NET

Framework .NET terdiri dari dua buah komponen utama, yaitu :

1. *Common Language Runtime (CLR)* adalah pondasi utama dari Framework .NET. CLR merupakan komponen yang bertangung jawab terhadap berbagai macam hal, seperti bertanggung jawab untuk melakukan manajemen memory, melakukan eksekusi kode, melakukan verifikasi terhadap keamanan kode, menentukan hak akses dari kode, melakukan kompilasi kode, dan berbagai layanan sistem lainnya. Dengan adanya fungsi CLR ini, maka aplikasi bebasis .NET biasa juga disebut dengan *managed code*, sedangkan aplikasi diluar itu biasa disebut dengan *unmanaged code*.

   Berikut ini beberapa hal yang disediakan CLR bagi para developer :

   a. Dapat lebih menyederhanakan proses pengembangan aplikasi.
   b. Memungkinkan adanya variasi dan integrasi dari berbagai bahasa pemrograman yang ada di lingkungan Framework .NET.
   c. Keamanan dengan melakukan identing pada kode aplikasi.
   d. Bersifat assembly pada saat proses *deployment* / kompilasi.
   e. Melakukan versioning sebuah komponen yang bisa didaur ulang.
   f. Memungkinkan penggunaan kembali kode, dengan adanya sifat *inheritance*.
   g. Melakukan pengaturan / manajemen tentang *lifetime* sebuah objek.
   h. Melakukan penganalisaan objek – objek secara otomatis.

Tugas utama CLR adalah mengkonversi kode MSIL (*Microsoft Intermediate Language*) menjadi *native code.* Proses kompilasi terjadi pada saat aplikasi tersebut dijalakankan, biasa di sebut JIT (*Just In Time*) *Compilation*. JIT

hanya akan mengkompilasi metode – metode yang memang digunakan dalam aplikasi, dan hasil kompilasi ini sendiri dichace didalam mesin dan akan dicompile kembali jika memang ada perubahan pada kode aplikasi kita.

2. *.NET Framework Class Library* atau sering juga disebut *Base Case Library* (BCL) adalah pustaka standar yang tersedia untuk semua bahasa pemrograman pada .NET. yang terdapat kumpulan tipe data dan kelas yang terintegrasikan dengan CLR. *Class library* bersifat berorientasi terhadap objek yang akan menyediakan tipe dari fungsi – fungsi *managed code*. Hal ini tidak hanya berpengaruh kepada kemudahan dalam hal penggunaan, tetapi juga dapat mengurangi waktu yang diperlukan pada saat eksekusi. Dengan adanya BCL ini, maka kita bisa menggunakan Framework .NET untuk membuat berbagai macam aplikasi, seperti :

   a. Aplikasi *console.*
   b. Aplikasi berbasis windows (windows form).
   c. Aplikasi ASP .NET (berbasis web).
   d. Aplikasi *web services* XML.
   e. Aplikasi berbasis *windows services.*

   Jika kita membuat sekumpulan *class* untuk membuat aplikasi berbasis windows, maka *class – class* itu bisa kita gunakan untuk jenis aplikasi lain, seperti aplikasi berbasis web (ASP .NET).

2.10.2.2 Keuntungan Framework .NET

Berikut beberapa keuntungan dari Framework .NET :

1. Mudah

   Kemudahan disini lebih kearah pada kemudahan bagi para developer untuk membuat aplikasi yang dijalankan pada lingkungan Framework .NET. beberapa hal yang merepotkan developer pada saat membuat aplikasi, telah dihilangkan atau diambil alih kemampuannya oleh Framework .NET, misalnya masalah *lifetime* sebuah objek yang biasanya luput dari perhatian developer pada saat proses pembuatan aplikasi.

2. Efisien

   Kemudahan pada saat proses pembuatan aplikasi, dan berimplikasi terhadap efisiensi dari suatu proses produktivitas, baik efisien dalam hal waktu pembuatan aplikasi atau juga efisien dalam hal lain, contohnya biaya (*cost*).

3. Produktivitas

   Semua kemudahan – kemudahan diatas, pada akhirnya akan membuat produktivitas menjadi lebih baik. Produktivitas naik, terutama produktivitas para developer, akan berdampak pada meningkatnya produktivitas suatu perusahaan.

4. Konsisten

   Kemudahan – kemudahan pada saat proses pembuatan aplikasi, juga bisa berimplikasi terhadap konsistensi pada aplikasi yang kita buat. Misalnya, dengan adanya BCL, maka kita bisa menggunakan objek atau class yang

dibuat untuk aplikasi berbasis windows pada aplikasi berbasis web. Dengan adanya kode yang bisa diintegrasikan ke dalam berbagai macam aplikasi ini, maka konsistensi kode – kode aplikasi kita dapat terjaga.

2.10.3 Bahasa C#

C# (dibaca C-*Sharp*) merupakan bahasa pemrograman perusahaan *Microsoft* yang didesain dengan target diimplementasikan dalam teknologi *Framework .NET*. Pada tahun 2002, tepatnya bulan Januari, C# maupun *Framework .NET* diselesaikan dan dapat diimplementasikan di kalangan industri. Bahasa pemrograman C# merupakan bahasa baru yang sangat handal dan konsisten serta membawa kesan bahasa pemrograman yang modern. C# pada dasarnya bukan bahasa sulit karena intinya mengambil dari berbagai bahasa pemrograman yang telah ada. Bahasa yang paling banyak digunakan dalam sintaksis C# adalah bahasa *Java*. Dimana komposisi bahasa C# adalah sebagai berikut:

- 70 % *JAVA*
- 10 % C++
- 5 % *VISUAL BASIC*
- 15 % BARU

Visual C# merupakan bagian dari teknologi Visual Studio *.NET* . Pada intinya, Visual C# merupakan bagian dari teknologi *.NET* yang dikembangkan untuk memenuhi kebutuhan industri akan keberadaan teknologi IT modern, yang mampu mengatasi sistem informasi [14].

2.10.4 ASP .NET

ASP .NET merupakan framework pemrograman yang berjalan diatas *Common Language Runtime* (CLR), biasanya digunakan diserver untuk membangun aplikasi berbasiskan web, ASP .NET memiliki keunggulan jika dibandingkan dengan model pemrograman web sebelumnya, antara lain adalah :

1. Sistem konfigurasi web sudah memiliki default yang sesuai dengan konfigurasi web server saat aplikasi web dibuat.
2. Fleksibiltas. Karena ASP .NET berbasiskan *Common Language Runtime*, fleksibilitas dan kemampuan yang dimiliki platform .NET akan tersedia pula bagi developer dalam membangun aplikasinya.
3. Performansi yang lebih baik. ASP .NET adalah suatu program hasil kompilasi *Common Language Runtime* (CLR) yang dijalankan diserver, tidak seperti ASP yang lebih dikategorikan sebagai bahasa interpreter, ASP .NET memiliki keunggulan seperti aplikasi lain yang berjalan dalam CLR, yaitu *early binding, just in time compilation*, optimasi program dan proses *caching* yang lebih baik. Hal ini meningkatkan performansi yang jauh lebih baik.
4. *Code Behind Pages* yang menyebabkan pembuatan aplikasi web menjadi lebih mudah karena mirip dengan aplikasi windows..
5. Kemudahan administrasi. ASP .NET memiliki sistem konfigurasi yang dapat digunakan untuk mengkonfigurasikan aplikasi web dan server. Informasi ini disimpan dalam bentuk text sehingga memudahkan dalam administrasi.

6. Security. ASP .NET selain dapat diintegrasikan dengan *windows authentication* juga dapat dikonfigurasi per aplikasi, hal ini menjamin aplikasi yang dibuat, aman.

2.10.5 Visual Studio Web Development 2010 Ultimate

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu 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 dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *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).

Kali ini penulis menggunakan Visual Studio 2010 Ultimate, sebagai perangkat lunak untuk mengembangkan aplikasi ini.

2.10.6 Sql Server 2008

Sql Server 2008 adalah salah satu produk terkini Microsoft yang menyajikan bukti nyata visi Microsoft dalam menyediakan platform basis data yang mendukung aplikasi dengan kebutuhan kualitas dan integritas data yang sangat kritis sekalipun dengan memperkecil biaya infrastruktur dan memungkinkan semua pengguna mengakses informasi secara lebih efesien. Platform ini menyediakan berbagai fitur yang memfokuskan pada:

1. Kepercayaan, memungkinkan organisasi untuk menjalankan aplikasi dengan kebutuhan data yang kritis dengan dukungan tingkat keamanan, kehandalan, dan skalabilitas yang tinggi.
2. Produktivitas, memungkinkan organisasi untuk mengurangi waktu dan biaya yang dibutuhkan untuk mengelola infrastruktur data.
3. Kualitas, memungkinkan penyampaian informasi yang sesuai dengan kondisi dan kebutuhan pengguna.

2.10.7 LINQ

Languange Integrated Query (LINQ) adalah seperangkat fitur yang diperkenalkan dalam Visual Studio 2008 yang memperluas kemampuan query yang kuat dengan sintaks bahasa C # dan Visual Basic. LINQ memperkenalkan standar, mudah-pola belajar untuk query dan update data, dan teknologi dapat diperluas untuk mendukung semua jenis berpotensi menyimpan data. Visual Studio mencakup majelis penyedia LINQ yang memungkinkan penggunaan LINQ dengan kumpulan NET Framework, database SQL Server, Datasets ADO.NET, dan dokumen XML. [15]