BAB II

LANDASAN TEORI

2.1 Profil Singkat SDN Griya Bumi Antapani 32 Bandung

Sekolah Dasar Negeri Griya Bumi Antapani 32 Bandung yang ber alamat di Jl. Tanjungsari VI No. 40, Antapani Wetan, Kec. Antapani, Bandung. merupakan salah satu sekolah dasar yang berada di kota bandung. Sekolah ini banyak berprestasi dalam bidang akademik atau pun non akademik antara lain Juara III Kejuaraan Unit Kesehatan Sekolah (UKS) Tingkat Kota Bandung Tahun 2013, Juara I Lomba Tingkat II (LT-II) Pramuka Putra dan Putri Tingkat Kota Bandung Tahun 2015. Sekolah dasar ini memiliki fasilitas ruangan yang cukup lengkap seperti Ruang Perpustakaan, Ruang UKS, Mushola, Tempat Bermain / Olah raga, WC Murid, WC Guru, Kantin Sekolah, Tempat Parkir Sepeda, Lapangan Upacara, Ruang Kesenian, Gudang dan Ruang Komputer.[1]

2.1.1 Visi, Misi, Tujuan Sekolah Dasar Negeri Griya Bumi Antapani 32 Bandung

Sekolah Dasar Negeri Griya Bumi Antapani 32 Bandung mempunyai visi, misi, dan tujuan yang merupakan target yang akan dicapai untuk kesuksesan sekolah dalam menerapkan pendidikan yang dapat membuat para siswa berprestasi. Berikut ini adalah Visi dari SDN Griya Bumi Antapani 32 Bandung yaitu Membentuk Peserta Didik Unggul Dalam Prestasi Berlandaskan Imtaq Sesuai Dengan Kemajuan Ipteq.[1] Sedangkan Misi dari SDN Griya Bumi Antapani 32 Bandung adalah : [1]

1. Melaksanakan kurikulum yang adaktif dan proaktif
2. Menciptakan suasana pembelajaran produktif, kreatif, inovatif, dan afektif (PIKEM) melalui pendekatan sentific.
3. Tercapainya lulusan yang unggul dalam prestasi akademik (Nilai UN sangat baik)
4. Mewujudkan peserta didik yang unggul dan mampu bersaing untuk melanjutkan ke SMP Negeri 80%.

5. Mewujudkan siswa unggul dalam prestasi lomba-lomba kreativitas siswa (juara kota)
6. Terbentuknya siswa yang berkarakter dengan perolehan nilai baik (B)
7. Melaksanakan kedisiplinan dan kewajiban sesuai dengan peraturan sekolah
8. Terciptanya lingkungan sekolah yang sehat, nyaman dan bersih melalui GPS
9. Mewujudkan pendidikan dan tenaga kependidikan yang memiliki kompetensi dan kualifikasi yang dipersyaratkan
10. Mewujudkan prasarana dan sarana pendidikan yang sesuai standar
11. Mewujudkan manajemen sekolah yang efektif dan efisien
12. Mewujudkan penggalangan biaya pendidikan yang gratis (sepenuhnya menggunakan alokasi dana bos)

Serta Tujuan yang ingin dicapai oleh pihak SDN Griya Bumi Antapani 32 Bandung antara lain :[1]

1. Mewujudkan Kurikulum yang adaptif
2. Mewujudkan Proses pembelajaran yang efektif dan efisien
3. Mewujudkan lulusan yang berakhlak mulia, cerdas, terampil, sehat jasmani dan rohani, kreatif, kompetitif, dan ramah terhadap lingkungan
4. Mewujudkan pendidik dan tenaga kependidikan yang memiliki kompetensi dan kualifikasi yang dipersyaratkan SNP
5. Mewujudkan prasarana dan sarana pendidikan yang sesuai standar yang diberlakukan
6. Mewujudkan manajemen sekolah yang efektif dan efisien
7. Mewujudkan penggalangan biaya pendidikan yang memadai
8. Mewujudkan standar penilaian prestasi akademik dan non akademik

2.2 Media Pembelajaran

Definisi media pembelajaran. Kata media merupakan bentuk jamak dari kata medium. Medium dapat didefinisikan sebagai perantara atau pengantar terjadinya komunikasi dari pengirim menuju penerima (Heinich et.al., 2002; Ibrahim, 1997; Ibrahim et.al., 2001). Media merupakan salah satu komponen komunikasi, yaitu sebagai pembawa pesan dari komunikator menuju komunikan (Criticos, 1996). Berdasarkan definisi tersebut, dapat dikatakan bahwa proses pembelajaran merupakan proses komunikasi. Proses pembelajaran mengandung lima komponen komunikasi, guru (komunikator), bahan pembelajaran, media pembelajaran, siswa (komunikan), dan tujuan pembelajaran. Jadi, Media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyalurkan pesan (bahan pembelajaran), sehingga dapat merangsang perhatian, minat, pikiran, dan perasaan siswa dalam kegiatan belajar untuk mencapai tujuan belajar.

2.2.1 Landasan Konsep Pembelajaran

1. Filsafat

   Secara filosofis belajar berarti mengingatkan kembali pada manusia mengenai makna hidup yang bisa dilalui melalui proses meniru, memahami, mengamati, merasakan, mengkaji, melakukan, dan meyakini suatu kebenaran sehingga semuanya memberikan kemudahan dalam mencapai segala yang di cita-citakan manusia. Belajar juga harus difahami sebagai sesuatu kegiatan dalam mencari dan membuktikan kebenaran.[3]

2. Psikologi

   Psikologi sebagai ilmu yang mempelajari gejala kejiwaan yang akhirnya mempelajari produk dari gejala kejiwaan ini dalam bentuk perilaku-perilaku yang tampak dan sangat dibutuhkan dalam proses belajar. Aliran psikologi ini sangat dominan dalam menentukan arah aktivitas manusia dalam melakukan proses pembelajaran. Dalam kegiatan pembelajaran seorang guru harus menguasai berbagai teori belajar, seperti teori belajar gestal, kognitif, dan humanistik. Hal ini digunakan karena

teori-teori belajar tersebut dijadikan landasan dalam mengembangkan kegiatan pembelajaran.[3]

3. Sosiologi

Kita sebagai makhluk sosial harus mempelajari lawan bersosialisasi, teman hidup bersama, dan mampu membangun masyarakat sampai dengan negara dan bangsa. Jika dalam proses pembelajaran tanpa arah dan tujuan pada makna hidup manusia sebagai makhluk sosial, maka belajar akan dijadikan cara untuk saling menguasai, memusnahkan, karena segala sesuatu yang dipelajari, diketahui, dipahami melalui belajar tidak digunakan dalam menciptakan kondisi kedamaian dunia.[3]

4. Komunikasi

Proses belajar atau pembelajaran akan menghasilkan suatu kondisi dimana individu dalam hal ini siswa dan guru, siswa dengan siswa atau interaksi yang kompleks sekalipun pasti akan ditemukan suatu proses komunikasi. Landasan komunikasi ini akan banyak memberikan warna dalam bentuk pendekatan, model, metode, dan strategi pembelajaran,serta pola-pola inovasi pembelajaran. Komunikasi cukup mampu memengaruhi peserta didik dalam mencapai keberhasilan membaca pesan-pesan atau informasi pembelajaran.[3]

5. Teknologi

Pembelajaran erat kaitannya dengan teknologi pendidikan, pembelajaran yang komprehensif harus memerhatikan perbedaan internet siswa, dimana siswa ada yang tipe auditif, visual dan kinestetik. Penggunaan teknologi dalam pembelajaran lebih akomodtif dan menyenangkan, sehingga pada gilirannya akan meningkatkan kualitas pembelajaran. Seorang guru pada proses pembelajaran melakukan kegiatan pembelajaran dapat menggunakan meedia pembelajaran, mulai dari yang sederhana seperti media gambar, foto, alat peraga, dan sampai ke media yang menggunakan teknologi canggih seperti LCD Projektor, penggunaan komputer dalam pembelajaran seperti e-learning, pembelajaran online, pembelajaran berbasis komputer.[3]

2.2.2 Pembelajaran Berbasis Komputer

Pembelajaran berbasis komputer menurut Wena, adalah pembelajaran yang menggunakan komputer sebagai alat bantu. Melalui pembelajaran ini bahan ajar melalui media komputer sehingga kegiatan pembelajaran lebih menarik dan membangkitkan gairah belajar siswa. Sedangkan menurut Hick dan Hyde (dalam Wena 2011) menyatakan bahwa dimana pada pembelajaran berbasis komputer siswa akan lebih mudah berinteraksi dan berhadapan langsung dengan teknologi secara individual sehingga siswa akan mendapatkan pengalaman yang membekas pada diri siswa. Dan menurut warsita menyatakan pembelajaran berbasis komputer adalah merupakan satu media pembelajaran yang sangat menarik dan mampu meningkatkan motivasi belajar peserta didik.[4]

2.2.3 Motion Tracking

Motion Tracking Adalah istilah yang digunakan untuk mendeskripsikan perekaman gerakan dan pengertian gerakan tersebut menjadi model digital. Motion tracking disimulasi sebagai alat analisis photogrammetric dalam penelitian biomechanics pada tahun 1970-an dan 1980-an, serta meluas ke ranah edukasi, latihan, olahraga, dan baru saja ke ranah animasi komputer untuk televisi, sinema, dan video games. Istilah Hand Motion Tracking merupakan salah satu istilah untuk salah satu implementasi teknik tracking and motion, dimana proses tracking dilakukan pada objek tangan manusia, Hand tracking dapat di implementasikan dalam banyak hal dan bidang, seperti dapat digunakan untuk mengenal bahasa isyarat, dan dapat dipakai menjadi teknik interaksi antara manusia dan komputer berbasis visi dan lain sebagainya. Tangan manusia adalah sebuah struktur mekanik kompleks yang terdiri atas beberapa segmen tulang, ligame-ligamen yang menghubungkan antar segmen tulang secara leluasa, otot-otot yang berperan sebagai motor gerak, tendon yang berperan untuk menghubungkan otot dengan tulang, dan kulit serta saraf-saraf halus yang menyelubungi otot dan tulang. Tulang-tulang saling terhubung pada pesendian dan tidak berubah ukurannya. Otot-otot menghasilkan tenaga penggerak dan menggerakkan sendi-sendi. Berdasarkan tipe gerakan dan rotasi yang mungkin dilakukan, sendi-sendi pada

tangan manusia dapat diklasifikasikan sebagai flexion, twist, directive, atau spherical. Contoh sendi bertipe flexion dengan 1 DOF adalah lutut dan siku, sedangkan contoh sendi bertipe twist dengan 1 DOF adalah sendi pronation dari lengan bawah. Gerakan direktif dari sendi dengan 2 DOF menghasilkan gerakan flexion dengan arah lebih dari 2. Sendi spherical, sebagaimana pada sendi bahu, memiliki 3 DOF dan dapat melakukan gerakan directive dan twist secara simultan [15].

2.2.4 Gesture Understanding

Interaksi dan komunikasi Manusia dengan mesin melalui penggunaan gesture dapat menjadi salah satu cara untuk meningkatkan kenyamanan dalam proses interaksi antara manusia dengan mesin, terlebih lagi bila model interaksi yang disajikan berbasis ruang 3 dimensi. *Gesture* understanding untuk model gerakan tangan dapat dikelompokkan menjadi 2 definisi umum, yang pertama gerakan *gesture* (tangan) sebagai representasi pergerakan tokoh dalam permainan, kedua adalah *gesture* sebagai pengontrol dan pengguna terhadap sebuah aplikasi / sistem. Sebagai contohnya ketika seseorang ingin memulai sebuah sistem atau aplikasi dengan menggunakan mouse atau keyborad, proses ini dapat digantikan dalam gerakan gesture yang memberikan arti yang sama, salah satu contohnya adalah penggunaan *Tapgesture, swipe gesture* dan lain sebagainya. Sehingga hal ini tergantung pada definisi yang diterapkan pada sebuah sistem. Model Interaksi gesture dapat dibagi menjadi dua kelompok, yakni :

a) Model interaksi gesture berdasarkan pola bentuk gesture, dimana sistem bekerja dengan menangkap gambar dan mengenali gesture dari identifikasi bentuk, salah satu contohnya pada gesture berdasarkan pola tangan dimana pendeteksian gerakan tangan dengan pola tangan yang ditangkap oleh leap motion controller sesuai dengan pola gerakan tangan pengguna.
b) Pola gerakan gesture, dimana sebuah gesture ditangkap dengan membaca sinyal yang dibangkitkan dari sensor yang telah terpasang, salah satu contohnya dideskripsikan pada gesture berdasarkan gerakan dimana

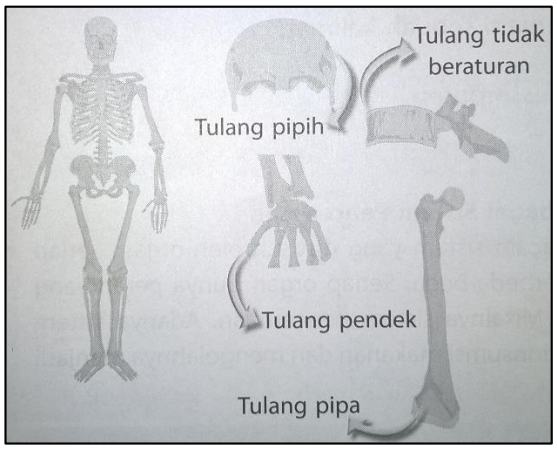
gerakan tangan pengguna dibaca berdasarkan arah dan pola gerakannya.[15]

2.3 Organ Tubuh Manusia dan Hewan

Tubuh makhluk hidup terdiri dari organ-organ. Organ adalah alat tubuh yang memiliki fungsi tertentu. Sistem organ adalah gabungan dari berbagai organ untuk melakukan fungsi tertentu di dalam tubuh. Setiap organ memegang peranan yang sama penting dalam menjalankan fungsinya.[5]

1. Bentuk Tulang Pada Manusia

   Berdasarkan bentuknya, tulang dibedakan menjadi empat, yaitu tulang pipa, tulang pipih, tulang pendek, dan tulang tak beraturan. Rangka manusia dibedakan menjadi rangka kepala, rangka badan, dan rangka anggota gerak. Rangka kepala dan rangka badan memiliki fungsi melindungi organ-organ tubuh penting seperti otak, paru-paru, jantung, atau hati. Rangka anggota gerak berfungsi untuk berjalan, berlari, atau memegang benda.[5]

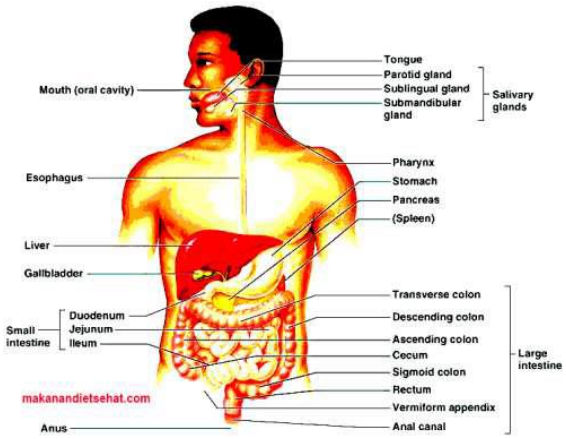


Gambar 2.1 Bentuk Tulang Pada Manusia[5]

2. Sistem Pencernaan Manusia

   Pencernaan adalah proses melumatkan makanan yang semula kasar menjadi halus. Makanan yang sudah di cerna, telah di ubah menjadi sari makanan dalam bentuk yang lebih halus sehingga mudah diserap oleh

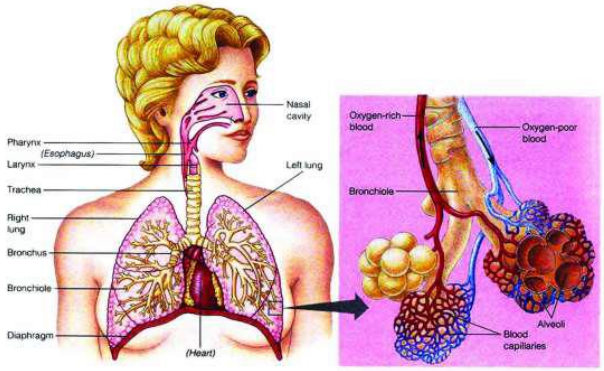
pembuluh darah. Kemudian oleh darah, sari makanan tersebut diedarkan keseluruh bagian tubuh. [6]



Gambar 2.2 Organ Sistem Pencernaan Manusia [7]

3. Sistem Pernapasan Manusia

   Sistem pernapasan pada manusia adalah proses menghirup udara dan mengeluarkan udara. Di dalam udara terkandung berbagai gas, salah satunya adalah oksigen. Gas yang dibutuhkan oleh tubuh manusia adalah oksigen. Oleh karena itu, hanya oksigen yang diambil dari udara untuk pernapasan. Pernapasan mengeluarkan gas karbon dioksida. Gas karbon dioksida dikeluarkan dari tubuh melalui hidung.[6]

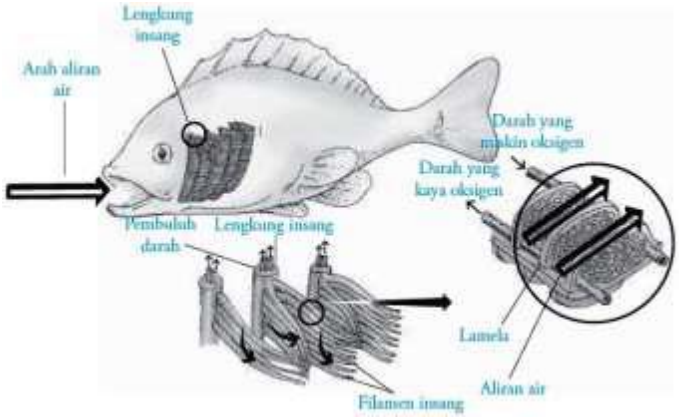


Gambar 2.3 Organ Sistem Pernapasan Manusia[7]

4. Sistem Pernapasan Hewan

   Organ pada sistem pernapasan hewan berbeda-beda. Ada hewan yang organ pernapasannya insang. Ada pula yang organ pernapasannya berupa

paru-paru, kulit, atau trakea. Perbedaan organ pernapasan pada hewan dipengaruhi oleh keadaan tubuh dan tempat hidupnya.[6]



Gambar 2.4 Organ Sistem Pernapasan Pada Ikan[7]

2.4 *Perceptual Computing*

Perceptual computing atau komputasi perseptual adalah sebuah metodologi yang membantu kita dalam melakukan penilaian subjektif. Lotfi Zadeh (1996, 1999, 2008), ayah dari logika *fuzzy*, menciptakan istilah "*Computing With Words*" (CWW) yang artinya komputasi melalui kata-kata sehingga objek dari komputasinya merupakan kata-kata yang diambil dari *natural language* atau bahasa natural. Ini terisnspirasi oleh kemampuan manusia yang luar biasa untuk melakukan berbagai tugas-tugas fisik dan mental tanpa pengukuran dan perhitungan. *CWW* mungkin memiliki pengaruh penting terhadap bagaimana manusia membuat keputusan yang rasional berdasarkan persepsi-dalam lingkungan ketidaktepatan, ketidakpastian, dan kebenaran parsial.[9]

Dengan kata lain *perceptual computing* mengubah cara interaksi manusia dengan komputer menggunakan bahasa natural seperti kata-kata, karena *perceptual computer (Per-C)* mampu membuat keputusan berdasarkan persepsi selayaknya manusia. *Natural language* atau bahasa natural yang kita kenal tidak hanya melalui kata-kata, bahasa tubuh atau gesture, maupun bahasa isyarat juga termasuk didalam *natural language* maka *perceptual computer (Per-C)* juga harus dapat mengenali bahasa tubuh, serta bahasa isyarat tersebut [9].

2.5 *Intel Perceptual Computing*

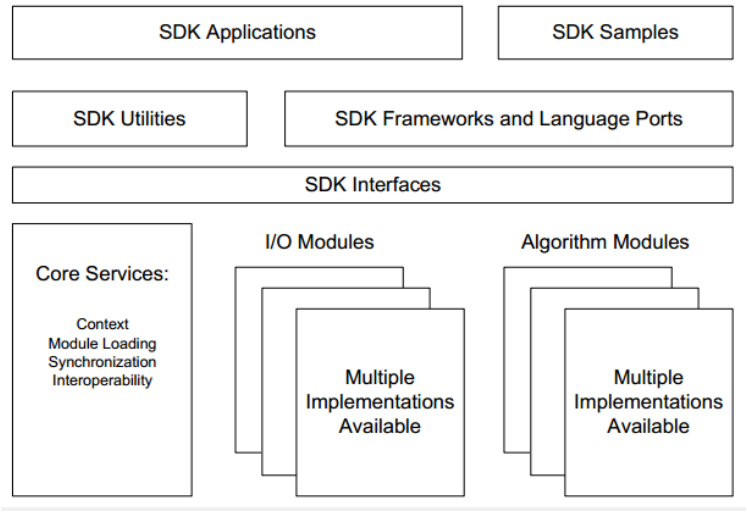
Intel Perceptual Computing merupakan produk Intel yang mengubah cara interaksi manusia dengan komputer dengan memanfaatkan pengenalan suara, pengenalan gerakan tangan, pengenalan wajah, bahkan *augmented reality*. Dengan kamera *Intel Perceptual Computing*, dan *Intel Perceptual Computing SDK* memungkinkan *developer* membangun perangkat lunak dengan fitur-fitur tersebut.[8]

Kamera *Intel Perceptual Computing* merupakan kamera *eksternal* yang dihubungkan melalui USB layaknya *webcam eksternal*, namun disertai sensor yang dapat melakukan pengenalan suara, gerakan tangan, pengenalan wajah, hingga *augmented reality*. [9]



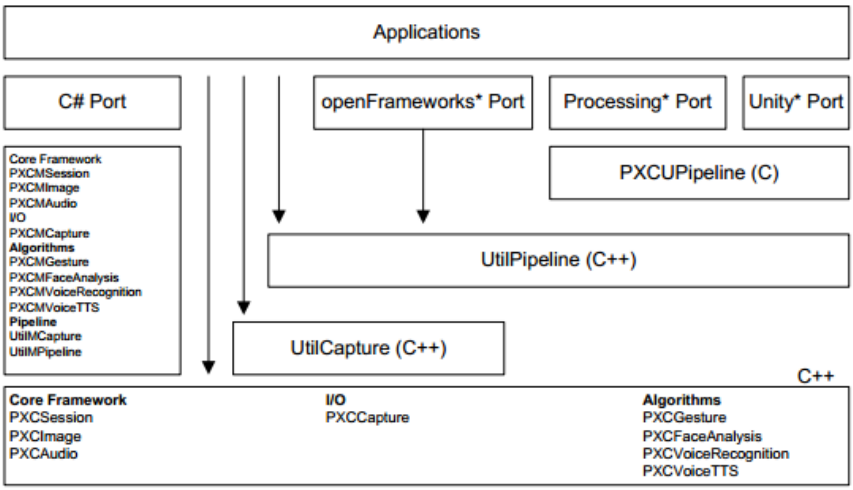
Gambar 2.5 Kamera *Intel Perceptual Computing* [8]

Intel Perceptual SDK adalah *library pattern detection* dan dan penerapan *recognition algorithm* yang terpapar melalui antarmuka yang sudah distandarkan. Tujuan *library* ini adalah untuk mengurangi hambatan pengembang untuk menggunakan algoritma ini dan mengalihkan fokus pengembang dari kode yang rinci menjadi lebih kearah berinovasi pada penggunaan algoritma ini untuk interaksi manusia dan komputer generasi berikutnya [8]



Gambar 2.6 Arsitektur *Intel Perceptual Computing SDK* [8]

Arsitektur dari *library SDK* seperti yang tertera pada gambar 2.6 terdiri dari beberapa lapisan komponen. Esensi dari fungsionalitas SDK terdapat pada modul I/O, dan modul algoritma. I/O modul berperan sebagai sumber retrieve data dari input device atau sink yang mengkonsumsi data ke *output device*. Modul algoritma termasuk didalamnya berbagai *pattern* detection dan algoritma *recognition algorithm* yang merupakan unsur penting dalam inovasi *human computer experience*, seperti *face recognition*, *gesture recognition*, *voice recognition*, dan *text to speech*. [8]



Gambar 2.7 Alur *Library Intel Perceptual Computing SDK* [8]

Seperti yang terlihat pada gambar 2.7 diatas, alur *library* yang membangun *SDK* adalah serangkaian dari *interface* C++, yang mendefinisikan fungsionalitas dari core framework, modul I/O, dan modul algoritma. *Interface* ini menyediakan *raw funcionalites*. Contohnya, *interface PXCCapture* mengambil sampel suara (*PXCAudio*), atau sampel gambar (*PXCImage*) dari sensor input. *Interface PXCCapture* melakukan *finger tracking*, dan *gesture recognition* sesuai dengan tipe input gambar sampel. [8]

SDK menyediakan beberapa kelas *utility* untuk menyederhanakan pemrograman aplikasi. Kelas *UtilCapture* merupakan *ekstensi* dari *interface PXCCapture* untuk membantu menghubungkan modul algoritma dengan *input device*, dan melakukan *stream* data. Kelas *UtilPipeline* menyediakan *interface* sederhana untuk penggunaan yang terbatas pada *gesture*, *face*, dan *voice algorithm*. [8]

2.5.1 Modul *Intel Perceptual Computing*

Kendali dalam aplikasi merupakan terapan dari dokumentasi *Intel Perceptual Computing SDK*, seperti *hand tracking*,. Berikut penjelasan dari modul

1. *Hand Tracking*

Sebelum membahas hand tracking, penting untuk memahami struktur dari GeoNode dan konsep pelabelan. Label mendefinisikan bagaimana body-related objects harus diidentifikasi dalam scene dan dibagi menjadi dua bidang utama, yaitu full-body label yang menyediakan informasi berkaitan dengan posisi tulang yang Nampak, dan hand-detailed label menyediakan informasi rinci dari objek tangan. Struktur GeoNode menjelaskan node geometris yang memberikan data relevan terkait objek yang sedang dilacak. Di baris kode di bawah ini, dengan asumsi bahwa pipeline telah dibuat dan diinisialisasikan. Kita menciptakan variabel body dan hand label diperlukan yang menentukan objek untuk dilacak. Dalam hal ini kita mencoba untuk melakukan recognize terhadap bagian tengah dari tangan. [9]

Dalam fungsi Update, kita mencoba untuk melakukan pengolahan acquire frame, untuk memastikan kita akan langsung keluar jika tidak ada acquired frame.

Jika sebuah acquire frame berhasil, kita memanggil fungsi QueryGeoNode. Fungsi QueryGeoNode mengembalikan rincian dari struktur GeoNode. Panggilan membutuhkan dua parameter, yaitu referensi untuk GeoNode yang struktur dimana hasil data akan disimpan, dan label tangan OR, yang menetapkan bahwa kita mencari bagian tengah dari tangan. Ketika permintaan untuk pusat-posisi tangan kembali benar, kita memiliki data yang valid disimpan dalam instance dari struktur GeoNode. Menggunakan data dari struktur GeoNode, kita dapat mulai menentukan interaksi pengguna dengan mendapatkan posisi tangan pengguna, kemudian menemukan dan memetakan aksi yang terjadi pada sistem yang kita buat. Setelah proses memperoleh hasil, kita diharuskan untuk melakukan release frame dan resources.[9]

```
        private PXCMGesture.GeoNode ndata;
        private PXCMGesture.GeoNode.Label bodyLabel =
        PXCMGesture.GeoNode.Label.LABEL_BODY_HAND_PRIMARY;
        private PXCMGesture.GeoNode.Label handLabel =
        PXCMGesture.GeoNode.Label.LABEL_HAND_MIDDLE;
        void Update()
        {
                if (!pp.AcquireFrame(false))
                return;
                if (pp.QueryGeoNode(bodyLabel | handLabel, out ndata))
        {
                //Get the standard hand position
                HandPosition.x = ndata.positionWorld.x;
                HandPosition.y = ndata.positionWorld.y;
                HandPosition.z = ndata.positionWorld.z;
        }
                pp.ReleaseFrame();
}
```

Gambar 2.8 Contoh Pemanggilan *HandTracking* [8]

2.6 Pemrograman Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespodensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas, sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan. Sistem berorientasi objek merupakan sebuah sistem yang komponennya dienkapsulasi menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya yang dapat berinteraksi satu sama lain [10]. Terdapat beberapa konsep utama pada metodologi berorientasi objek, diantaranya [11]:

1. Kelas (*class*), kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dari himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*), dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.
2. Objek (*object*), abstraksi sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, datau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.
3. Abstraksi (*abstraction*), prinsip untuk merepresntasikan dunia nyata yang kompleks menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

4. Enkapsulasi (*encapsulation*), pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan impelemntasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.
5. Pewarisan (*inheritance*), mekanisme yang memugkinkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.
6. Polimorfisme (*polymorphism*), kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.7 *Unified Modeling Language*

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*)[10]. 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.[11]

Terdapat beberapa diagram yang umum digunakan dalam pemodelan UML diantaranya [11]:

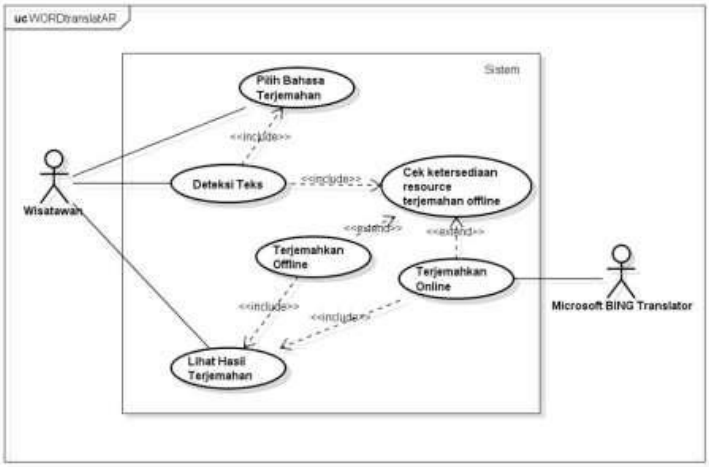
1. *Use Case Diagram*, merupakan pemodelan untuk kelakuan (behavior) sistem yang akan dibuat. Use case mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.
2. *Class Diagram*, menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki

atribut dan metode (operasi). Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode (operasi) fungsi-fungsi yang dimiliki oleh suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki kelas main, kelas view, kelas controller, serta kelas model.

3. *Sequence Diagram*, menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui onjek-objek yang terlibat dalam sebuah use cse beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case.
4. *Activity Diagram*, menggambarkan aliran kerja (workflow) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, melainkan aktivitas yang dapat dilakukan oleh sistem.

2.8 *Use Case Diagram*

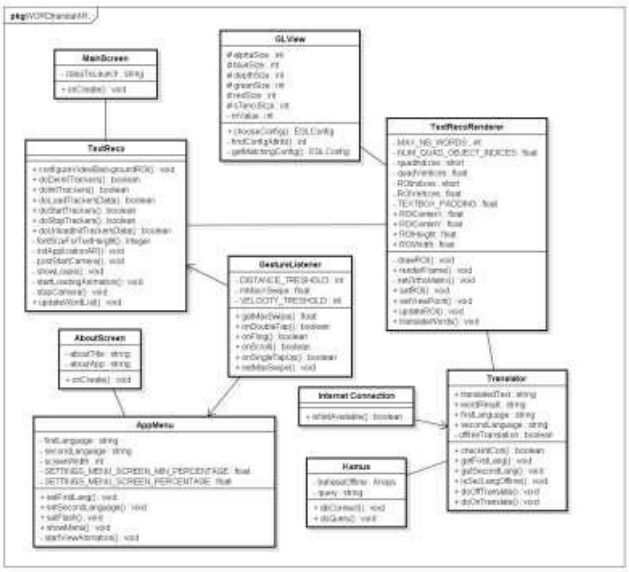
Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. Use Case Diagram dapat digunakan selama proses analisis untuk menangkap requirement sistem dan untuk memahami bagaimana sistem bekerja seperti pada gambar 2.9.[10]



Gambar 2.9. *Use Case Diagram*.[10]

2.9 *Class Diagram*

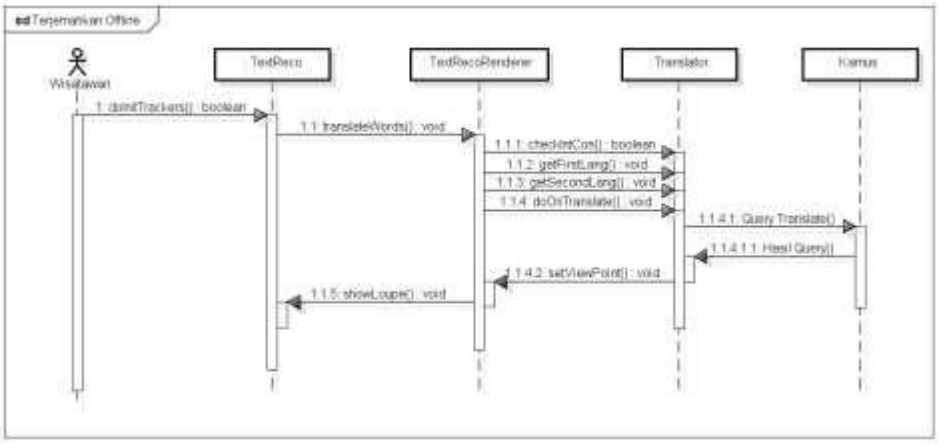
Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Class Diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, class diagram berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat seperti pada gambar 2.10. [10]



Gambar 2.10. ***Class Diagram***.[10]

2.10 *Sequnce Case Diagram*

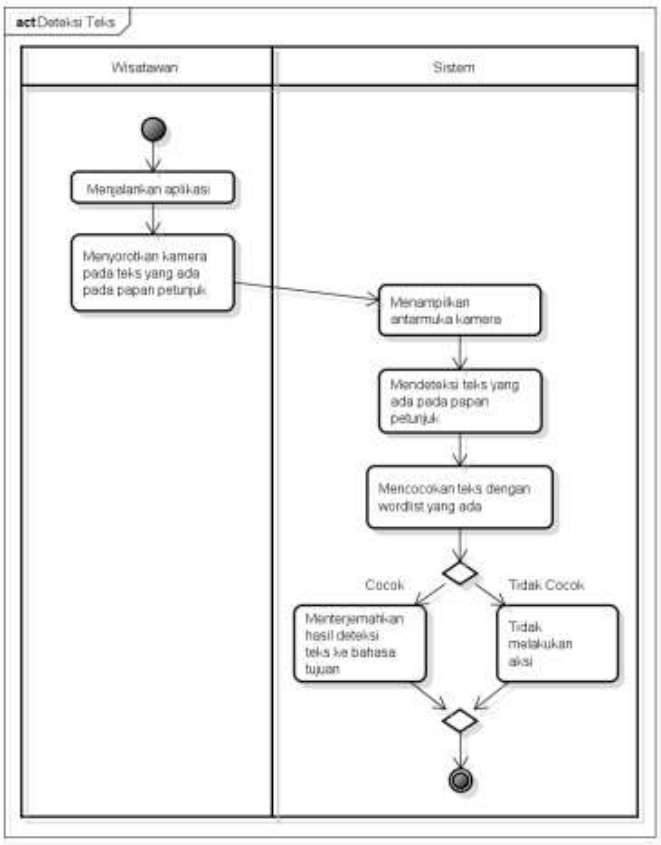
Menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem seperti pada gambar 2.11.[10]



Gambar 2.11. ***Sequence Diagram***.[10]

2.11 *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti use case atau interaksi seperti pada



Gambar 2.12 *Activity Diagram*.[10]

2.12 Bahasa Pemrograman C Sharp (C#)

Pada tahun 2000, Microsoft mengumumkan bahasa pemrograman C# (C Sharp) yang diciptakan secara khusus untuk platform .Net (dot NET). C# dikembangkan oleh Microsoft dengan merekrut Anders Helsberg. Tujuan dibangunnya C# adalah sebagai bahasa pemrograman utama dalam lingkungan .NET Framework. C# memiliki akar dari C, C++, dan Java. Seperti Visual Basic, C# berorientasi objek dan memiliki akses terhadap pustaka kelas .NET Framework Class Library, yaitu kumpulan yang kaya akan komponen-komponen terpradefinisi yang memungkinkan untuk mengembangkan aplikasi secara cepat.

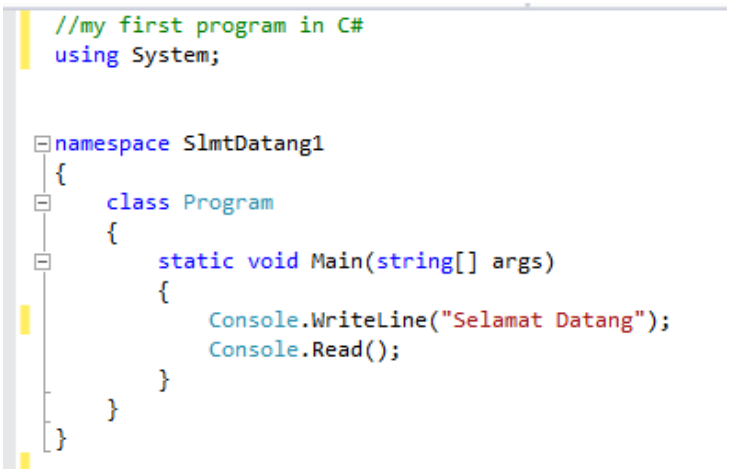
C# memiliki kapabilitas yang sama dengan Java dan cocok untuk aplikasi desktop, bergerak (mobile), dan aplikasi berbasis web.[13]

C# merupakan bahasa event-driven yang dapat merespon event yang dipicu oleh pengguna, seperti klik mouse, ketikan pada keyboard, pewaktu, dan lainnya. Visual C# merupakan bahasa pemrograman visual. Bahasa visual ini memberikan beberapa pernyataan program untuk membangun beberapa bagian dari aplikasi. Kemampuan GUI (graphical user interface) dari visual studio yang memungkinkan pengguna untuk melakukan drag and drop objek-objek terpradefinisi seperti tombol dan kotak teks ke tempat-tempat pada layar.[13]

2.12.1 Penulisan

Dalam menulis suatu program C# ada lima hal penting struktur penulisan dasar yaitu *Resource atau library*, Struktur pertama ini merupakan pendefinisian library apa yang harus ada pada program atau library apa yang di impor. *Namespace*, Struktur kedua ini adalah nama dari project program yang dibuat. *Name Class*, Struktur ketiga ini berbicara tentang apa nama dari *Class* yang dibuat dan bisa juga langsung diberi penanda seperti *Main Class* yang menandakan bahwa *Class* tersebut adalah *Class utama*. Deklarasi *Method*, Struktur keempat ini merupakan pendeklarasian *method* sebagai awalan untuk menjalankan *method* atau perintah yang ada di dalamnya, jika didefinisikan dengan "*Main*" maka *method* tersebut yang dijalankan pertama kali oleh *compiler*. *Method* atau *Command*, Struktur kelima adalah *method* atau perintah yang diberikan untuk di eksekusi oleh *compiler*. Bila ada salah satu struktur yang tidak ada maka kemungkinan besar terjadi error pada program, namun untuk struktur kedua biasanya setiap IDE akan otomatis.[13]

Program Bahasa C# tidak mengenal aturan penulisan di kolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C# diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C# seperti pada gambar 2.13 :[14]



```
//my first program in C#
using System;


namespace SlmtDatang1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Selamat Datang");
            Console.Read();
        }
    }
}
```

Gambar 2.13. *Source Code C Sharp.*

Gambar 2.13 merupakan source code, yang diberi nama SlmtDatang1.cs. Program diatas merupakan salah satu program paling sederhana dalam C# tetapi dalam program tersebut mengandung komponen dasar yang selalu ada pada setiap pemrograman. C# Jika dilihat satu persatu :

// my first program in C#

Baris ini adalah komentar. semua baris yang diawali dengan dua garis miring (//) akan dianggap sebagai komentar dan tidak akan berpengaruh terhadap program. Dapat digunakan oleh programmer untuk menyertakan penjelasan singkat atau observasi yang terkait dengan program tersebut.[13]

using System;

Intruksi baris *using System* merupakan salah satu contoh *library* atau *refrences* dari bahasa pemrograman C#. *Refrences* atau *library* mempunyai kumpulan-kumpulan fungsi tertentu. Contohnya fungsi dalam '*system'* adalah "*Console.WriteLine();*" dan "*Console.Read();*" tanpa adanya *refrences* tersebut jika menuliskan sebuah ekspresi atau fungsi *Console.WriteLine();* dan *Console.Read();* program tidak akan berjalan.[13]

namespace SlmtDatang1

Intruksi pada baris *namespace* BelajarCsharp merupakan sebuah deklarasi dimana dalam *namespace* tersebut "BelajarCsharp"

adalah nama dari deklarasi *name project*. *namespace* sendiri merupakan sebuah kumpulan atau koleksi kelas. ketika membuat sebuah *project* menggunakan *Visual Studio* deklarasi *namespace* akan memasukkan *name project*, dan *name project* tersebutlah yang akan digunakan sebagai deklarasi *namespace*.[13]

class Program

Intruksi pada baris *class Program* merupakan sebuah deklarasi sebuah *class*. *Class* adalah perluasan konsep dari struktural data. Serupa dengan struktural data, *class* dapat berisi anggota data (atribut), tetapi *class* juga dapat berisikan fungsi-fungsi sebagai bagian dari anggotanya.[13]

static void main(string[] args)

Intruksi pada baris *static void main(string[] args)* merupakan sebuah *main method* atau yang dikenal sebagai fungsi utama dalam pemrograman. Dimana ketika program dieksekusi yang pertama kali baca adalah "*static void main(string[] args)*" setelah itu baru dilanjutkan kepembacaan fungsi-fungsi lainnya yang disertakan dalam fungsi utama tersebut.[13]

Console.WriteLine("Saya sedang belajar Csharp");

Intruksi pada baris *Console.WriteLine("Selamat Datang");* merupakan sebuah ekspresi dalam *main method* atau fungsi utama. Dimana *Console.WriteLine();* berfungsi untuk menampilkan *output* pada *Console* atau Aplikasi.[13]

Console.Read();

Intruksi pada baris *Console.Read();* berfungsi untuk menahan *Console* agar ketika program dijalankan *console* akan tetap muncul, sampai menutup program dengan sendiri.[13]

2.12.2 Tipe Data

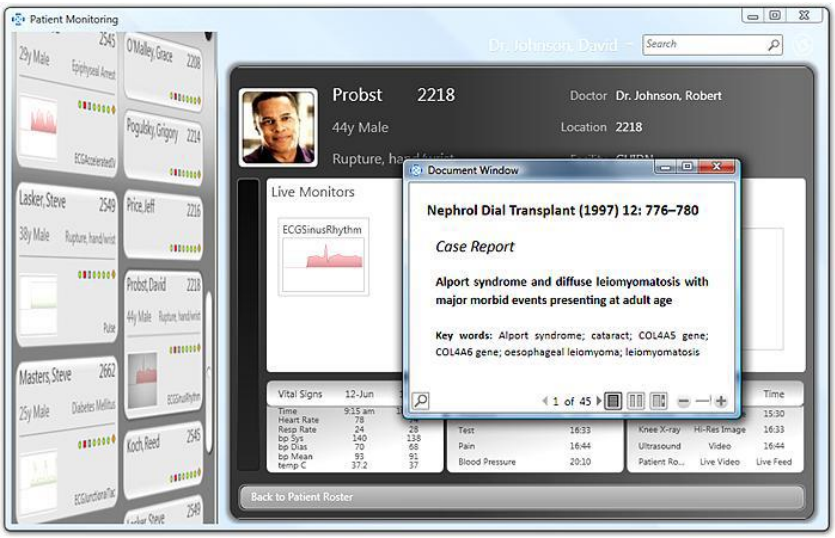
Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.[14]

Tabel 2.1. Bentuk Tipe Data.[2]

No.	Tipe Data	Range (Jangkauan)	Nilai Default	Keterangan
1	*bool*	True atau False	False	Boolean
2	*byte*	0 to 255	0	8-bit unsigned integer
3	*char*	U +0000 to U +ffff	'\0'	16-bit Unicode character
4	*decimal*	(-7.9 x 1028 to 7.9 x 1028) / 100 to 28	0.0M	128-bit precise decimal values with 28-29 significant digits
5	*double*	(+/-)5.0 x 10-324 to (+/-)1.7 x 10308	0.0D	64-bit double-precision floating point type
6	*float*	-3.4 x 1038 to + 3.4 x 1038	0.0F	32-bit single-precision floating point type
7	*int*	-2,147,483,648 to 2,147,483,647	0	32-bit signed integer type
8	*long*	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0L	64-bit signed integer type
9	*sbyte*	-128 to 127	0	8-bit signed integer type
10	*short*	-32,768 to 32,767	0	16-bit signed integer type
11	*uint*	0 to 4,294,967,295	0	32-bit unsigned integer type
12	*ulong*	0 to 18,446,744,073,709,551,615	0	64-bit unsigned integer type
13	*ushort*	0 to 65,535	0	16-bit unsigned integer type

2.13 Windows Presentation Foundation

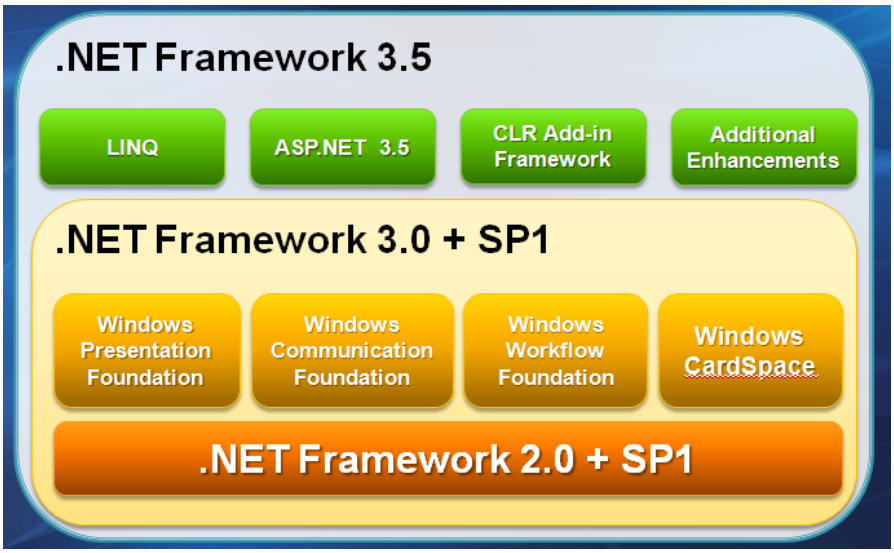
Windows Presentation Foundation (WPF) adalah sebuah sistem presentasi generasi berikutnya untuk membangun aplikasi-aplikasi klien *Windows* dengan *user experiences* yang mengagumkan secara visual. Dengan WPF, Anda dapat menciptakan berbagai aplikasi baik yang stAndalone maupun yang dihost oleh browser. Beberapa contohnya adalah Yahoo! Messenger dan New York Times Reader, seperti halnya juga Contoso Healthcare Sample Application yang dapat diperoleh di http://windowsclient.net yang diperlihatkan pada gambar 2.14 berikut.



Gambar 2.14. ***UI untuk Conoso Healthcare Application***.[12]

Inti dari WPF adalah sebuah mesin render yang berbasis vektor dan tidak bergantung pada resolusi yang dibangun untuk memanfaatkan *hardware* grafis modern. WPF memperluas intinya dengan sebuah set fitur pengembangan aplikasi yang menyeluruh yang mencakup *Extensible Application Markup Language* (XAML), kontrol, data *binding*, *layout*, grafik 2-D dan 3-D, animasi, *styles*, *templates*, dokumen, media, teks, dan tipografi. WPF tercakup didalam *Microsoft .NET Framework 3.0* atau yang lebih baru, sehingga dapat membangun aplikasi-aplikasi yang menyertakan aplikasi-aplikasi lain dari pustaka kelas .NET *Framework*. *Windows Presentation Foundation* (WPF) adalah sebuah subsistem grafis didalam *.NET Framework 3.0* (atau yang lebih baru) dan berkaitan secara langsung dengan bahasa aplikasi XAML. WPF tercakup dalam *Windows Vista*

dan *Windows Server 2008*, dan juga tersedia untuk *Windows XP Service Pack 2* atau yang lebih tinggi, dan *Windows Server 2003*. WPF menyediakan sebuah model pemrograman yang konsisten untuk membangun aplikasi dan menyediakan pemisah yang jelas antara antarmuka pengguna dan logika bisnis. Sebuah aplikasi WPF dapat dipasang pada komputer desktop atau di-host didalam sebuah *web browser*. Teknologi ini juga memungkinkan kontrol, desain, dan pengembangan aspek-aspek visual dari program-program Windows. WPF mencoba untuk menyatukan host layanan aplikasi: antarmuka pengguna, gambar 2D dan 3D, dokumen-dokumen tetap dan adaptif, tipografi tingkat lanjut, diagram vektor, diagram raster, animasi, pengikatan data, audio dan video. Walaupun *WinForms* akan terus digunakan secara luas, dan Microsoft hanya menciptakan sedikit aplikasi WPF, banyak berbagai studi kasus mempromosikan WPF untuk aplikasi-aplikasi lini bisnis.[12]



Gambar 2.15. WPF didalam .NET Framework 3.5. [12]

2.13.1 Fitur-fitur WPF

Fitur-fitur yang terdapat pada *Windows Presentation Foundation* (WPF) cukup lengkap. Dalam hal ini para pengembang perangkat lunak dapat mengkombinasikan antara design dengan proses pengkodean program pada saat sedang melakukan pengembangan perangkat lunak tampilan dari segi grafis menjadi lebih bagus dibanding dengan hanya menggunakan bahasa pemrograman C# saja. Fitur-fitur yang terdapat didalam WPF antara lain yaitu :

2.13.1.1 Antarmuka Tampilan Grafis

Semua grafis (termasuk item desktop seperti *windows*) adalah aplikasi *Direct3D*. *Direct3D* bukan bermakna bahwa semua tampilannya harus berantarmuka tiga dimensi. Pada bagian ini WPF melakukan enkapsulasi *Direct3D* dengan memberikan berbagai fitur di antaraya.

1. Memberikan sebuah jalan yang terpadu untuk menampilkan grafis, seperti halnya lebih banyak fitur grafis tingkat lanjut.
2. Merutekan grafis melalui *Direct3D* memungkinkan Windows mengalihkan beberapa tugas grafis ke *Graphics Processing Unit* yang ditemukan pada kartu grafis komputer. Hal ini dapat mereduksi beban kerja pada unit pemrosesan Central dari komputer.
3. Mendukung grafis berbasis vektor, yang memungkinkan penskalaan tanpa mengurangi kualitas.
4. Mendukung render dan interaksi model 3D dalam aplikasi-aplikasi 2D.
5. Konten 2D yang interaktif dapat diletakkan diatas permukaan 3D, secara natif.

2.13.1.2 Interoperabilitas

WPF memberikan interoperabilitas dengan *Win32*: Melalui konsep *hosting*, seseorang dapat menggunakan *Windows Presentation Foundation* didalam kode *Win32* yang ada, atau seseorang dapat menggunakan kode *Win32* yang ada didalam *Windows Presentation Foundation.* Interoperabilitas dengan *Windows Forms* juga mungkin dilakukan melalui penggunaan kelas *ElementHost* dan kelas *WindowsFormsHost.*

2.13.1.3 Layanan Media

WPF memberikan bangun primitive untuk grafis 2D bersama dengan seperangkat kuas, pena, geometri, dan transformasi yang built-in. Kemampuan 3D didalam WPF merupakan sebuah subset dari seperangkat fitur lengkap yang disediakan oleh *Direct3D*. Meskipun demikian, WPF memberikan integrasi yang lebih ketat dengan fitur-fitur lain seperti *user interface* (UI), dokumen, dan media. Hal ini memungkinkan untuk memiliki UI 3D, dokumen 3D, dan media 3D. Pada WPF Terdapat dukungan untuk sebagian besar format gambar umum yang telah distandarisasi. WPF mendukung juga format video *WMV, MPEG* dan beberapa berkas *AVI.* Pada pendekatan ini WPF juga mendukung animasi berbasis waktu, sebagai lawan dari pendekatan berbasis frame. Hal ini mengurangi kecepatan animasi dari bagaimana sistem bekerja.

2.13.1.4 Data Binding

WPF memiliki seperangkat layanan data yang built-in untuk memungkinkan pengembang aplikasi untuk mengikatkan dan memanipulasi data didalam aplikasi. Terdapat dukungan untuk tiga jenis data binding:

1. *One time*: dimana klien membiarkan updates pada *server*.
2. *One way*: dimana klien memiliki akses *read-only* ke data.
3. *Two way*: dimana klien dapat membaca dari dan menulis data ke *server*.

Pengikatan data tidak memiliki penghalang pada presentasinya. WPF memberikan template data untuk mengontrol presentasi data.

2.13.1.5 Antarmuka Pengguna

Seperangkat kontrol yang built-in diberikan sebagai bagian dari WPF. Kontrol ini memuat berbagai item-item seperti tombol, menu, dan *list box*. Hal yang mungkin dinotifikasi adalah salah satu kontrol WPF yangtidak memiliki kontrol penampil data berupa *DataGrid*, namun demikian vendor-vendor pihak ketiga telah menawarkan beberapa kontrol tersebut, atau bahkan dapat memperoleh kontrol tersebut tanpa investasi. Hal yang menarik dan diterapkan pada konsep pemograman .NET adalah diterapkannya konsep pemisah logis antara sebuah kontrol dari penampilannya.

1. Sebuah *template* dari kontrol dapat ditimpa untuk benar-benar mengubah tampilan visualnya.
2. Sebuah kontrol dapat memuat kontrol atau *layout* lainnya, yang memungkinkan kontrol yang belum pernah dilakukan diatas komposisi.
3. Fitur-fitur penyangga mode grafis, sehingga aplikasi-aplikasi tersebut tidak perlu terganggu dengan penggambaran tampilan kembali.

2.13.1.6 Anotasi

Anotasi adalah kemampuan member informasi penandaan pada dokumen. Hal ini dapat diterapkan pada basis per-obyek, untuk obyek-obyek didalam sebuah *Document* atau *FlowDocument*. WPF hanya memberikan kemampuan untuk menciptakan, menyimpan dan mengelola anotasi; masing-masing aplikasi harus mengekspos perancanfan antarmuka sendiri.

2.13.1.7 Imaging

WPF secara natif dapat mengakses pustaka standar *Windows Imaging Component* (WIC) dan API yang memungkinkan pengembang untuk menuliskan kodek gambar untuk format berkas gambar yang telah distandarisasi.

2.13.1.8 Efek

WPF dapat menghadirkan efek-efek *bitmap*, efek-efek ini ditampilkan dengan mengopimalkan seluruh kemampuan hardware yang menjalankan *.NET Framework 3.5 SP1* atau yang lebih baru. Fitur-fitur GPU seperti pixel shaders tidak digunakan untuk efek-efek bitmap. Efek-efek khusus seperti *dropshadows* dan *blurring* tersedia secara *built in* di teknologi *frameworknya*.

2.13.1.9 Dokumen

WPF secara natif mendukung dokumen-dokumen layaknya seperti dokumen word. WPF menyediakan sebuah kelas yang diberi *DocumentReader*, yang digunakan untuk membaca dokumen-dokumen dengan *layout* tetap. Kelas *FlowDocumentReader* menawarkan mode-mode tampilan yang berbeda seperti per-halaman atau dapat di-skrol dan juga menyusun kembali teks jika area tampilan diubah ukurannya. Konsep halaman-halaman pada dokumen WPF ini mengikuti standar XML *Paper Specification*, serta mendukung pembacaan dan penulisan dokumen-dokumen berhalaman menggunakan *Open Packaging Convention*.

2.13.1.10 Teks

WPF mencakup sejumlah fitur penampilan teks dan tipografi yang sebelumnya tidak tersedia didalam GDI. Teknologi WPF ini menghadirkan antarmuka pemrograman Windows yang pertama untuk mengekspos fitur-fitur *OpenType*. *OpenType* adalah teknologi skalabilitas suatu ukuran font pada layar komputer. WPF menangani teks dalam standar *Unicode*, dan menangani independesi teks dari sisi pengaturan global dari sistem Windows. Pada pengelolaan teks terdapat fitur-fitur menarik yakni.

1. mekanisme *fallback* yang memungkinkan arah tulisan dapat bersifat horizontal dan vertikal.
2. Mengembangkan font internasional yang bersifat komposit dari *font-font* yang sudah ada.
3. Informasi font komposit yang ditata rapih dalam suatu skema xml

Pengelolaan tekas pada WPF juga mendukung pemeriksaan ejaan yang *built-in*. Pengelolaan teks WPF ini mendukung fitur-fitur seperti spasi otomatis, teks internasional berbahasa inggris, pemisahan baris, penyambungan kaya, perataan paragraf, efek-efek *bitmap*, tranformasi *font*, dan efek-efek teks ala animasi seperti bayangan, samar (*blur*), *glow*, rotasi, dsb. Berbagai macam efek tersebut memungkinkan pengembangan teks beranimasi yang bersifat *real-time*.

2.13.1.11 Input Alternatif

WPF mendukung fungsionalitas yang terkait dengan tinta digital. Input tinta digital ini mengadopsi dukungan teknologi *Windows Ink Services Platform* (WISP) yang sudah dikenal pada implementasi tablet PC di tahun 2005 silam. Hal yang menarik dari WISP ini adalah mengkombinasikan dan mengenkapsulasi teknologi input alternatif yang ditanamkan saat ini pada *Windows 7*.

2.13.1.12 Aksesabilitas

WPF mendukung *Microsoft UI Automation* yang memungkinkan para pengembang untuk menciptakan antarmuka yang dapat diakses dengan mudah, bagi para pengguna yang mengalami aksebilitas terbatas.

2.13.2 WPF Programming Model (XAML)

XAML adalah salah satu bahasa implementasi yang mengikuti permodelan keberhasilan bahasa markup untuk pengembangan web. WPF memperkenalkan sebuah bahasa baru yang dikenal sebagai *eXtensible Application Markup Language* (XAML), yang berdasarkan pada XML. XAML dirancang sebagai sebuah metode yang lebih efisien untuk mengembangkan antarmuka pengguna aplikasi. Keuntungan khusus yang dibawa XAML bagi WPF adalah bahasa yang dihasilkan menggunakan mekanisme deklaratif. Pada bahasa pemrograman yang deklaratif, pengembang (atau perancang) mendeskripsikan perilaku dan integrasi dari komponen-komponen tanpa penggunaan pemrograman prosedural. Hal ini memungkinkan bagi seseorang dengan sedikit atau tanpa pengalaman pemrograman untuk membuat seluruh aplikasi kerja tanpa pemrograman.

Walaupun secara umum jarang sekali ditemukan sebuah aplikasi yang dibangun secara menyeluruh dalam XAML, pengenalan XAML mengijinkan perancang aplikasi untuk berkontribusi secara lebih efektif pada siklus pengembangan aplikasi. Penggunaan XAML untuk mengembangkan antarmuka pengguna juga mengijinkan pemisahan model dan tampilan, dan hal ini merupakan sebuah prinsip arsitektural yang baik. Pada XAML, elemen-elemen dan atribut-atribut dipetakan ke kelas-kelas dan properti-properti dalam API dasar. Seperti di dalam pengembangan web, baik *layout* maupun *theme* yang khusus dapat di-*markup* untuk aplikasi WPF, namun XAML tidaklah dibutuhkan secara penuh. Semua elemen WPF dapat dikodekan dalam bahasa .NET (C#, VB.net). Kode XAML pada akhirnya dapat dikompilasi kedalam sebuah kode terkelola dengan cara yang sama dengan semua bahasa .NET. Pada point ini meyakinkan bahwa penggunaan XAML untuk pengembangan tidak menambah beban performa. XAML juga dapat dikompilasi dan dijalankan secara *"on demand"* mirip dengan halaman *web* HTML. Walaupun XAML telah diperkenalkan sebagai sebuah bagian integral dari WPF, standar XAML sendiri tidak menjadi khusus bagi WPF (atau bahkan .NET). XAML juga dapat digunakan untuk mengembangkan aplikasi-aplikasi menggunakan berbagai API pemrograman lain dan tidak bergantung pada bahasa tertentu. Meskipun demikian, penelitian khusus telah dilakukan dalam mengembangkan API WPF untuk memaksimalkan interoperabilitas dengan model deklaratif yang diperkenalkan oleh XAML. Banyak aplikasi, seperti *Microsoft PowerPoint* dan *Word*, akan mendukung *ekspor* konten mereka ke XAML. Ada beberapa subset atau profil dari XAML, seperti:

1. XAML Presentation (XAML-P) – menyertakan semua item yang memuat WPF v1.0.
2. XML Paper Specification (XPS) – sebuah subset dari XAML-P untuk merepresentasikan dokumen-dokumen format yang fix dan digunakan sebagai format spool untuk subsistem pencetakan didalam *Windows 7*.

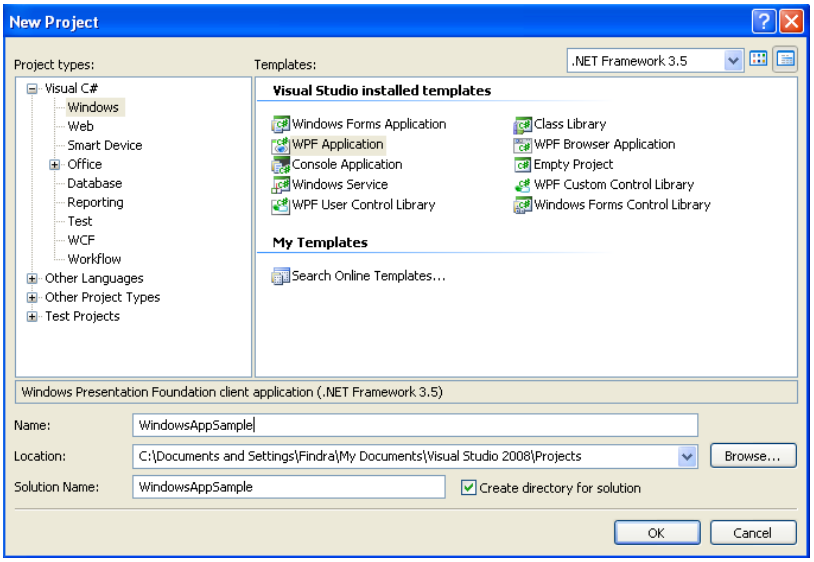
2.13.3 Dasar Aplikasi WPF

Windows Presentation Foundation (WPF) merupakan generasi penerus setelah *Windows Forms* untuk pengembangan aplikasi desktop. Perbedaannya dengan aplikasi *Windows Forms* tradisional, aplikasi WPF memisahkan kode untuk UI dari kode untuk fungsionalitas aplikasi. WPF adalah sebuah kemajuan yang besar dalam pengembangan aplikasi desktop. WPF tidak hanya mengijinkan kontrol UI melalui penggunaan grafis tingkat lanjut, namun juga memungkinkan pengembangan yang lebih cepat karena pemisahan kode antara presentation layer dan business logic layer mengijinkan pengembangan yang terpisah antara desainer dan developer.

2.13.4 Windows Applications

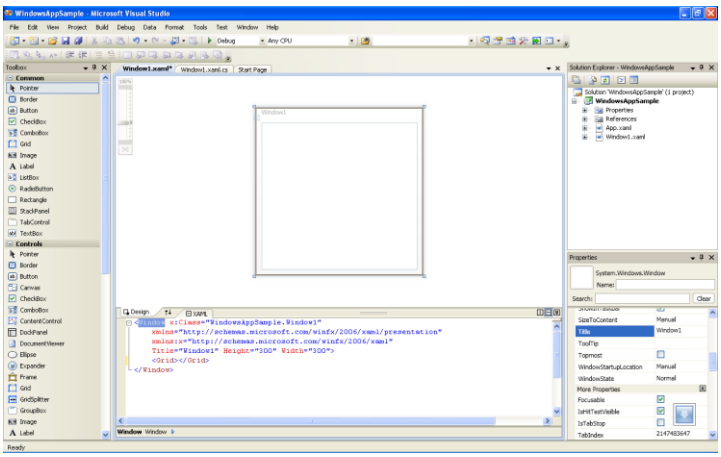
Windows applications mirip dengan aplikasi *Windows Forms*. *Windows applications* memberikan sebuah *user experience* yang familiar bagi pengguna dan pengembang *Windows*. Pada jenis aplikasi ini dapat dibuka banyak jendela pada satu waktu, namun tidak terdapat sistem navigasi atau histori. Windows application terdiri dari satu atau lebih jendela dan *business logic* yang terkait. Elemen UI level teratas dari sebuah jendela adalah sebuah kelas yang diturunkan dari kelas *Window*. Kelas *Window* diturunkan dari *ContentControl*, yang berarti bahwa ia dapat memuat sebuah elemen tunggal didalam properti *Content*-nya. Elemen ini biasanya merupakan sebuah kontrol layout yang dapat memuat banyak kontrol anak. Untuk menciptakan sebuah Windows application baru, caranya adalah:

1. Didalam *Visual Studio*, pilih menu *File* > *New* > *Project*.
2. Terbuka kotak dialog *New Project*.
3. Didalam kotak dialog *New Project*, pilih *WPF Application*, berikan sebuah nama (misalnya *WindowsAppSample*), dan klik OK.



Gambar 2.16 Kotak dialog New Project. [12]

Sebuah aplikasi baru tercipta didalam *Visual Studio integrated development environment* (IDE), seperti yang ditampilkan dalam Gambar 1.2. Didalam *Visual Studio* IDE untuk aplikasi *WPF*, *Toolbox*, *Properties*, *Designer*, dan *Solution Explorer* sangat mirip dengan *Windows Forms designer*. Yang baru disini adalah XAML *view* yang terletak dibawah designer. XAML *view* mengijinkan untuk mengedit XAML yang mendefinisikan jendela didalam designer.

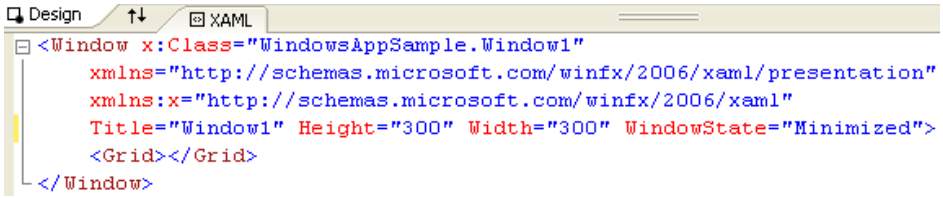


Gambar 2.17 Sebuah Windows application baru didalam Visual Studio IDE.[12]

Penetapan nilai properti *Window* dapat dilakukan dengan salah satu dari dua cara berikut ini:

1. Pada saat desain dengan memilih jendela didalam designer dan menetapkan properti yang sesuai didalam jendela Properties.
2. Mengedit XAML untuk jendela tersebut didalam XAML *view*.

Contoh berikut mendemonstrasikan pengaturan properti *WindowsState* (dalam kotak hijau) dengan mengaturnya didalam XAML untuk jendela:



Gambar 2.18 Pengubahan properti Window pada XAML View [12]

2.14 *Visual Studio* 2012

Microsoft *Visual Studio* 2012 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.[13]

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*).[13]



Gambar 2.19 Logo Visual Studio 2012.[13]

2.15 Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam angket dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti ini : [19]

a. Sangat tidak setuju
b. Tidak setuju
c. Netral (biasa saja)
d. Setuju
e. Sangat setuju

Selain pilihan dengan lima skala seperti contoh di atas, kadang digunakan juga skala dengan tujuh atau sembilan tingkat. Suatu studi empiris menemukan bahwa beberapa karakteristik statistik hasil kuesioner dengan berbagai jumlah pilihan tersebut ternyata sangat mirip.[20]

Adapun rumus perhitungan skala likert adalah sebagai berikut,

Indeks	Nilai Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Sering	2
Sangat Tidak	1

Perhitungan total nilai, Perhitungan nilai tertinggi dan terendah, Nilai indeks hasil 10 / 5 = 2, dan didapatkan interval sebagai berikut.

Tabel 2. 1 Perhitungan Rumus Likert

Perhitungan	Rumus
Total nilai	banyak kusioner x nilai indeks
Nilai Tertinggi	nilai indeks ter tinggi x jumlah
Nilai Indeks Hasil	100 nilai skor tertinggi

2.16 *Business Process Model and Notation* (BPMN)

Business process model and notation (BPMN) adalah Representasi grafis untuk menentukan proses bisnis dalam suatu pemodelan proses Bisnis. Tujuan utama dari BPMN adalah Menyediakan suatu notasi standar yang mudah dipahami oleh semua pemangku kepentingan bisnis, mulai dari bisnis analis yang membuat draft awal dari proses, para pengembang teknis yang bertanggung jawab untuk menerapkan teknologi yang akan melakukan proses-proses tersebut, hingga kepada orang-orang bisnis yang akan mengelola dan memantau proses mereka. Kategori dasar dari elemen BPMN adalah :

1. *Flow Objects*

Flow object adalah elemen utama yang menggambarkan karakteristik dari sebuah proses bisnis. *Flow object* terdiri dari 3 elemen yaitu :

1. *Event* : *Event* adalah sesuatu yang terjadi selama rangkaian dari sebuah *business process*. *Event* ini mempengaruhi alur dari proses dan biasanya

memiliki sebuah tindakan atau sebuah hasil. Simbolnya adalah bentuk lingkaran dan berdasarkan alur pengaruhnya, terdapat 3 jenis *event* yaitu:

1) *Start event* : *Start Event* ini adalah simbol yang mengindikasikan sebuah proses yang akan dimulai
2) *Intermediate event* : Simbol Intermediate terletak diantara *Start Event* dan *End Event*. Simbol Ini akan mempengaruhi alur proses, tetapi tidak akan memulai atau secara langsung menghentikan proses.
3) *End event* : Di indikasikan sebagai simbol untuk mengakhiri sebuah proses.

2. *Activities* : *Activities* digunakan untuk mewakili berbagai makna dalam kehidupan sehari-hari. Aktivitas dianggap mencakup berbagai kegiatan yang dapat diselesaikan dalam waktu 5 menit, satu minggu atau lebih.
3. *Gateway* : Dalam BPMN, *Gateway* dapat mendefinisikan semua tindakan Arus Urutan Proses Bisnis. Sebuah *Gateway* kadang-kadang memainkan salah satu dari dua peran, dan kadang-kadang bermain baik pada waktu yang sama.

2. *Connecting Objects*

Connecting Object adalah elemen yang menghubungkan *Flow object*. *Connecting Object* juga memiliki 3 jenis elemen yaitu :

1) Alur *Sequence* (*Sequence flow*) digunakan untuk menunjukkan urutan yang kegiatan akan yang dilakukan dalam sebuah proses.
2) Alur Pesan (*Messege Flow*) digunakan untuk menunjukkan aliran pesan antara dua entitas yang siap untuk mengirim dan menerima.
3) Asosiasi (*Association*) digunakan untuk asosiasi data, informasi dengan aliran benda

3. *Swimlanes*

Swimlanes adalah pengelompokan dari beberapa model elemen. *Swimlanes* digunakan untuk memisahkan dan mengatur kegiatan oleh peserta sehingga kita dapat memahami siapa yang bertanggung jawab untuk setiap event. *Swimlanes* ada 2 jenis yaitu :

1) *Pools* : Semua Bisnis Flow Diagram mengandung setidaknya satu Pool
2) *Lanes* : Pool yang dibagi lagi menjadi sub *Swimlane*.

4. *Artifacts*

Artifacts adalah elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses. Bentuk dan penggunaan artifacts itu bermacam-macam dan bisa lebih luas tergantung dari standar pengertian BPMN yang digunakan. Saat ini terdapat 3 setingan pengaturan artifacts, yaitu :

1) Data *Object* : mekanisme untuk menunjukan bagaimana data dibutuhkan atau diproduksi oleh aktivitas. Data *object* dihubungkan dengan aktivitas melalui *Associations.*
2) *Group* : diwakili dengan persegi panjang dengan ujung bulat yang digambarkan dengan garis putus-putus. *Group* dapat digunakan untuk tujuan dokumentasi atau analisis, tetapi tidak mempengaruhi *Sequence Flow.*
3) *Annotations* : mekanisme untuk pemodel memberikan informasi *teks* tambahan untuk pembaca dari diagram BPMN.