BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Profil perusahaan merupakan pembahasan mengenai sejarah, visi dan misi, struktur organisasi serta deskripsi pekerjaan pada perusahaan yang akan menjadi tempat dibangunnya system ini.

2.1.1 Sejarah PT. Azizi Audhinia Wisata

PT.AZIZI AUDHINIA WISATA Berdiri pada tahun 1998 dan dilegalitaskan pada tahun 2005 Oleh Hj. Naslah Lubis, yang bergerak dibidang jasa travelling, yaitu suatu usaha jasa yang Membantu perjalanan baik untuk keperluan usaha/bisnis maupun wisata.

Dengan menyediakan penawaran jasa sebagai berikut:

1. Ticketing (Domestic & Internasional)
2. Land Arrangment Paket Wisata
3. Hotel (Domestik & Internasional)
4. Dokumen Travel (Visa,Paspor,dan KITAS)
5. Paket Tour (Domestik &Internasional)

2.1.2 Visi dan Misi

Pencapaian suatu tujuan diperlukan suatu perencanaan dan tindakan nyata untuk dapat mewujudkannya, secara umum bisa dikatakan bahwa visi tindakan dengan apa yang direncanakan untuk mencapai suatu tujuan. Berikut dibawah ini adalah penjelasan Visi Misi PT. Azizi Audhinia Wisata :

1. Visi

    a. Menjadi salah satu biro perjalanan wisata terbaik di Indonesia
    b. Berkontribusi dalam meningkatkan industry parawisata dan transportasi/perjalanan di Indonesia

2. Misi

    a. Selalu menghadirkan produk-produk dan layanan terbaik.
    b. Memberikan kemudahan dan kenyamanan bagi setiap pelanggan.
    c. Secara terus menerus meningkatkan kemampuan SDM dan infrastruktur perusahaan dapat memberikan pelayanan terbaik bagi setiap pelanggannya.
    d. Memberikan edukasi melalui paket wisata yang sudah dikemas secara menarik dan ekslusif.

2.1.3 Logo PT. Azizi Audhinia Wisata

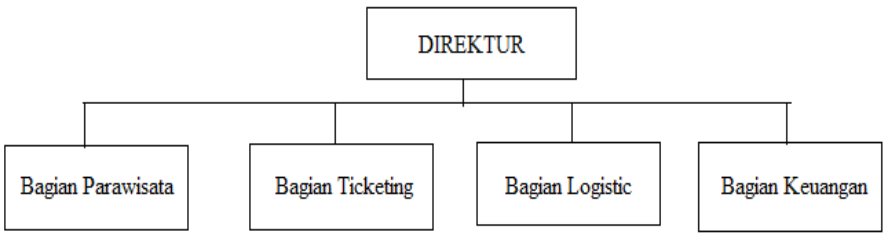
Logo perusahaan merupakan identitas suatu perusahaan yang membedakan perusahaan satu dengan yang lainnya. Berikut merupakan logo PT. Azizi Audhinia Wisata dapat dilihat pada gambar 2.1.



Gambar 2.1 Logo Perusahaan

2.1.4 Struktur Organisasi PT. Azizi Audhinia Wisata

Struktur organisasi merupakan hal yang sudah pasti ada dalam perusahaan agar pegawainya dapat bekerja sesuai dengan *job description* masing-masing. Berikut merupakan struktur dan *job description* PT. Azizi Audhinia Wisata.



Gambar 2.2 Struktur Organisasi PT. Azizi Audhinia Wisata.

2.1.5 Deskripsi Pekerjaan

Deskripsi kerja struktur organisasi dapat dilihat pada penjelasan berikut :

a. Direktur bertugas sebagai penanggung jawab segala keputusan perusahaan.
b. Bagian. Parawisata bertugas dengan pengurusan data pelanggan beserta kelengkapan dokumen-dokumen, serta bertugas sebagai marketing melayani pemesanan dan memasarkan program perusahaan, serta menyampaikan segala kebutuhan informasi untuk program parawisata kepada pelanggan.
c. Bagian. Ticketing Berkenan sebagai pengurusan ticket wisata.
d. Bagian. Logistic mengurus kelengkapan untuk para wisatawan.
e. Bagian. Keuangan Berkenan sebagai pengurus administrasi pelanggan.

2.2 Landasan Teori

Landasan teori dari penulisan skripsi ini menguraikan proses analisis system serta mendukung proses perancangan System *customer relationship management* di PT. Azizi Audhinia Wisata.

2.2.1 *Customer Relationship Management (CRM)*

Manajemen hubungan pelanggan (CRM- *Customer Relationship Management)* adalah proses mengelola informasi rinci tentang pelanggan perorangan dan semua “titik kontak” pelanggan secara seksama untuk memaksimalkan loyalitas pelanggan.

Pengertian CRM adalah manajemen hubungan pelanggan yaitu suatu jenis manajemen yang secara khusus membahas teori mengenai penanganan hubungan antara perusahaan dengan pelanggannya dengan tujuan meningkatkan nilai perusahaan di mata pelanggannya. CRM melingkupi semua aspek yang berhubungan dengan calon pelanggan dan pelanggan saat ini. Lebih lanjut *Customer Relationship Management* adalah strategi tingkat korporasi yang berfokus pada pembangunan dan pemeliharaan hubungan dengan pelanggan, jadi CRM lebih menekankan pada pendekatan holistik terhadap falsafah organisasi yang menekankan hubungan yang erat dengan pelanggan.[3]

2.2.1.1 Tujuan Dan Manfaat CRM

Tujuan pembuatan (CRM) Customer Relationship Management adalah

1. agar dapat menciptakan hubungan yang lebih baik antara perusahaan dengan customer sehingga pihak perusahaan dapat mengenal lebih jelas karakteristik dari pelanggan yang beranekaragam
2. memberikan masukan dan pendapat tentang bagaimana cara penerapan CRM (Customer Reletionship Managemant) di dalam suatu organisasi dan juga faktor-faktor maupun kiat-kiat yang harus dilakukan ataupun dipenuhi dalam penerapan CRM Meningkatkan Efisiensi Operasional

Manfaat pembuatan (CRM) Customer Relationship Management adalah

1. menjadikan keberadaan marketing, customer support dan customer care menjadi lebih efektif
2. diharapkan para pelanggan bisa merasakan pelayanan yang memuaskan dari perusahaan, dan diharapkan pula perusahaan akan terbiasa dengan melihat kondisi para pelanggannya terlebih dahulu sebelum para pemimpin perusahaan mengambil sebuah keputusan.
3. Pembaca dapat mengerti tentang faktor-faktor kunci kesuksesan dalam penerapan CRM
4. Menambah wawasan pembaca tentang cara yang harus dilakukan agar terhindar dari kegagalan penerapan CRM[9]

2.2.1.2 Mengimplementasikan CRM

Customer Relationship Management adalah strategi tingkat korporasi, yang focus pada pembangunan dan pemeliharaan hubungan dengan pelanggan. Beberapa paket perangkat lunak telah tersedia dengan pendekatan yang berbeda-beda terhadap CRM. Bagaimanapun, CRM bukanlah teknologi itu sendiri, tapi ia adalah pendekatan holistik terhadap falsafah organisasi, yang menekankan hubungan yang erat dengan pelanggan. CRM mencakup filosofi organisasi pada semua tingkatan, termasuk kebijakan dan proses, *customer service*, pelatihan pegawai, pemasaran, dana manajemen sistem dan informasi. Sistem CRM

mengintegrasikan pemasaran, penjualan, dan *customer service* dari ujung ke ujung.[3]

2.2.1.3 Jenis-Jenis CRM

Secara garis besar, CRM dapat diklasifikasikan ke dalam 3 jenis, yaitu:

1. Operasional CRM

Mencakup otomatisasi yang terintegrasi dari keseluruhan proses bisnis yang terintegrasi, termasuk didalamnya customer touch point dan customer channels dan integrasi front office dan back office. Operasional CRM mengotomatisasi proses bisnis yang berinteraksi secara langsung dengan pelanggannya. CRM memiliki berbagai aplikasi yang diterapkan dalam penjualan, pemasaran, dan pelayanan yang mendukung proses bisnis. Aplikasi yang mendukung proses bisnis di atas tergolong dalam aplikasi operasional. Aplikasi operasional CRM dibedakan menjadi dua, antara lain: (1) customer facing applications, faktor utama dari aplikasi crm customer facing adalah contact center, Sales Force Automation, dan Field Service. Disebut Customer Facing karena pada kenyataannya, aplikasi ini langsung berhubungan dengan pelanggannya; (2) customer touching applications, faktor utama dari aplikasi CRM Customer-touching applications adalah campaign management, E-commerce dan self service customer support disebut customer touching karena pelanggan berhubungan langsung dengan aplikasi sistem daripada berhubungan dengan perwakilan perusahaan. Strategi operasional CRM ini memfasilitasi tradisional CRM, yang berfokus bagaimana membuat pelayanan, penjualan atau departemen pemasaran
lebih efisien dan meningkatkan profibilitas perusahaan dan pelanggan.

2. Analytic CRM

Merupakan proses analisis dari data-data yang dihasilkan pada Operational CRM. Tantangan bagi perusahaan sekarang ini adalah untuk mengerti apa yang menjadi permintaan pelanggan dan memberikan respon, secara lebih baik, mengantisipasi kebutuhan mereka. Namun, banyak perusahaan hanya berfokus untuk mengimplementasikan penggunaan CRM hanya sebatas untuk ‘mendengar’

kebutuhan dan keluhan dari pelanggan, yang pada praktisnya hanya berkonsentrasi pada komponen *operational* dan *collaborative* CRM, sehingga analisis akan pemahaman dan pengenalan terhadap pelanggan yang sebenarnya belum terjadi. Diperlukan komponen *Analytical* CRM untuk mengoptimalkan hubungan perusahaan dengan pelanggannya.

Analytical CRM memungkinkan perusahaan untuk mendapatkan suatu pengetahuan akan pelanggan dan melakukan penaksiran atau estimasi terhadap pelanggan berdasarkan data-data analisis yang dipakai. Pengetahuan yang didapat perusahaan adalah tentang nilai pelanggan, yang dapat mendukung chanel pelayanan interaksi dengan lebih baik dan mendukung berbagai keputusan dalam mensinergi penerapan operational dan atau *collaborative* CRM dalam proses penjualan dan pemasaran produk, sehingga keudukannya menjadi sangat esensial bagi component CRM lainnya.

Analytical CRM merupakan *feedback loop* antara interaksi pelanggan yang *realtime* yang terjadi pada *front-end* atau *back-end Scorecard* untuk menganalisa apa yang telah terjadi dan bagaimana meningkatkan cara perusahaan berinteraksi dikemudian hari. Mengatakan bahwa *analytical* CRM adalah penggunaan data pelanggan untuk menganalisis, pemodelan dan evaluasi yang ditujukan untuk mendukung perusahaan untuk membentuk suatu hubungan yang *profitable* antara perusahaan dan pelanggannya.

Analytical CRM terdiri dari semua *programming* yang menganalisis data tentang pelanggan perusahaan, sehingga keputusan yang lebih cepat dan lebih baik dapat dihasilkan.

Berdasarkan sebuah arkitel dalam Info World Analytical CRM dapat menyediakan hal-hal sebagai berikut :

1. kelompok segmentasi pelanggan
2. Analisa profitabilitas, mengetahui pelanggan mana yang *profitable* selama kurun waktu tertentu

3. personalisasi, kemampuan untuk memasarkan kepada pelanggan secara personal berdasarkan data-data yang ada tentang pelanggan itu.

4. *event monitoring*, yaitu segala aktivitas yang dilakukan pelanggan yang berkaitan dengan perusahaan dan begitu pula sebaliknya tindakan perusahaan terhadap segala yang berhubungan dengan pelanggannya, seperti promosi, transaksi, dan sebagainya.

3. Kolaboratif CRM

Aplikasi pelayanan yang terkolaborasi, seperti email, personalized publishing, e-communities, forum diskusi dan sarana lainnya yang dirancang untuk memfasilitasi interaksi antar pelanggan dan pihak perusahaan. CRM yang terdiri dari ketiga komponen di atas memiliki tujuan secara umum yaitu untuk memaksimumkan keuntungan pelanggan sejalan dengan meningkatkan dan memelihara kepuasan pelanggan.[9]

2.2.2 *Sms Gateaway*

SMS Gateway merupakan sebuah perangkat yang menawarkan layanan pengiriman pesan ke jaringan selular dari media lain, atau sebaliknya, sehingga memungkinkan pengiriman atau penerimaan pesan dengan atau tanpa menggunakan ponsel [6]. SMS Gateway dapat terhubung ke media lain seperti perangkat SMSC dan server milik Content Provider melalui link IP untuk memproses suatu layanan SMS. Sebuah sistem SMS Gateway, umumnya terdiri komponen hardware (server/komputer yang dilengkapi dengan perangkat jaringan) dan software (Aplikasi yang digunakan untuk pengolahan pesan). Dan untuk sebuah sistem yang besar umumnya dapat menggunakan database untuk penyimpanan data.[4]

2.2.2.1 *SMS (Short Message Service)*

SMS adalah protokol telekomunikasi yang membantu kita mengirimkan pesan pendek (sebanyak 160 karakter) berupa karakter alfanumerik [6]. SMS merupakan layanan komplementer dari 2 layanan utama sistem jaringan nirkabel

digital, baik GSM (Global System for Mobile Communication) maupun CDMA (Code Division Multiple Access), yakni layanan voice dan switched data.[4]

2.2.3 Metode RFM

RFM, yakni metode yang berdasar pada Recency, Frequency, dan Monetary. Recency adalah mengukur nilai pelanggan dengan melihat perilakunya yang berkenaan dengan pembelian yang dilakukan paling akhir. Setiap kali pelanggan melakukan pembelian yakinkan bahwa semua infornasi yang berkenaan dengan pembelian terakhir itu diperbaharui, baik secara otomatis maupun manual. Informasi terpenting yang tidak boleh terlewatkan berkenaan dengan recency adalah tanggal pembelian terakhir yang merupakan barometer pengukuran recency. Frequency adalah frekuensi menggunakan dasar seberapa sering sesorang melakukan pembelian dalam waktu tertentu. Monetary adalah berpatokan pada nilai uang atau nilai transaksi yang dibelanjakan oleh pelanggan terhadap produk atau jasa yang kita tawarkan pada jangka waktu tertentu[5]. Formula R/F/M bukanlah penemuan baru. R/F/M telah menjadi standar bagi perusahaan yang menjual merchandise sejak setengah abad lampau. Pada awalnya formula ini dikembangkan oleh George cullinan untuk perusahaan yang bernama Alden's. formula tersebut kemudian diikuti oleh banyak perusahaan lain, termasuk raksasa supermarket di amerika seperti 'Sears Roebuck' dan 'Montgomery Ward' Formula RFM tidak harus selalu sama antar satu pemasar dengan pemasar lainnya, tergantunng pada prioritasnya terhadap 3 (tiga) variable berikut:

Pada kondisi tertentu, mungkin juga dibutuhkan pemberian bobot yang lebih besar, misalnya kepada pelanggan yang membeli paling akhir. Tetapi pemasar lain mungkin menganggap frekuensi pembelianlah yang harus diberi bobot yang lebih tinggi. Alasannya frekuensi lebih baik bila digunakan untuk menggambarkan tingkat loyalitas pelanggan di banding recency dan monetary. Bagaimana pembobotan yang dilakukan, sepenuhnya tergantung pemasar yang bersangkutan. Tidak ada kombinasi yang paling benar atau salah, lagi pula, pada tahap imokementasinya nanti, kombinasi pembobotan ini dapat diseuaikan kembali sesuai dengan kebutuhan waktu itu. Selanjutnya pemasar harus menentukan poin

masing-masing variable untuk beberapa kondisi yang berbeda. Asumsi yang mendasari keputusan pemberian poin pada recency, misalnya didasarkan pada semakin baru ia berbelanja, maka semakin tinggi poin yang didapatkan. RFM terdiri dari dari dua metode, yakni: mensortir data-data pelanggan berdasarkan RFM dan menganalisa hasilnya (Recency, Frequency dan monetary) masing-masing punya kategori-kategori tertentu, dan dari irisan antara atribut dan kategori, terciptalah segmen-segmen. Kemudian metode ini menetapkan skor bagi tiap pelanggan berdasarkan data recency,frequency dan monetary mereka. Sehingga tiap pelanggan mempunyai weight points pada masing-masing atribut, yang kemudian jika disatukan dengan hasil dari atribut lainnya menjadi poin kumulatif. Melalui kedua analisa RFM ini, maka Anda akan dapat memahami perilaku konsumen secara lebih mendalam, sehingga dapat membantu Anda dalam melakukan segmentasi dan targeting selanjutnya. Anda dapat mengetahui segmen-segmen bahkan individual mana saja yang paling berharga bagi perusahaan. Artinya, untuk melakukan RFM ini Anda membutuhkan data yang mendetail mengenai pelanggan.[6] keterangan point pelanggan dapat dilihat pada tabel 2.1.

Tabel 2.1 Keterangan Point Customer

Score	R	F	M
1	20<25 bulan	1-5	500rb < 3jt
2	15<20 bulan	6-13	3jt<6jt
3	10<15 bulan	14-17	6jt<20jt
4	5<10 bulan	18-23	20jt<50jt
5	0<5 bulan	24-100	50>100jt

Skala yang diberikan untuk masing-masing variabel recency, frequency dan monetary adalah 1 sampai 5. Nilai 1 diberikan untuk pelanggan yang memiliki recency,frequency, dan monetary pembelian yang sedikit di antara pelanggan lain. Nilai 5 diberikan untuk pelanggan yang mempunyai frequency, recency, dan *monetary* pembelian lebih sering daripada pelanggan lain. Setelah pelanggan diberikan nilai score variabel, selanjutnya menghitung total score masing-masing pelanggan dengan bobot yang sama yaitu 10. Tentu saja perusahaan dapat

melakukan analisa atau kajian secara lebih mendalam dengan mengembangkan konsep RFM ini menjadi dimensi-dimensi kecil lainnya. Penggunaan metode statistik semacam mean, standard deviation, variance, mode, median, dan lain sebagainya juga dapat turut berperan dalam usaha untuk membagi pelanggan menjadi beberapa segmen

2.2.4 Model Analisis

Model analisis yang digunakan untuk membangun adalah sebagai berikut:

2.2.4.1 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa pemodelan standar yang memiliki sintaks dan semantic. Pemodelan ini sangat cocok digunakan untuk merancang dan memodelkan sistem berorientasi objek[9].

Diagram pada UML dibagi menjadi dua bagian yaitu *structural* diagram dan *behavior* diagram. *Structural* diagram digunakan untuk mendeskripsikan relasi antar kelas. *Tools* yang digunakan pada bagian ini yaitu *class diagram*. Sedangkan *behavior* diagram digunakan untuk mendeskripsikan interaksi antara aktor dan sebuah *use case* (bagaimana seorang aktor menggunakan sistem). *Tools* yang digunakan pada bagian ini yaitu *Use case* diagram, *Sequence* diagram, *Collaboration* diagram, *Statechart* diagram, dan *Activity* diagram.

2.2.4.2 *Diagram Class*

Diagram *class* digunakan untuk menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem[9]. Sebuah *class* terdiri dari nama, atribut dan *method.* Atribut dan *method* dari sebuah kelas mempunyai visibility, Ada tiga jenis visibility yang digunakan yaitu *private, public,* dan *protected*. Setiap *class* pada *class* diagram mempunyai hubungan dengan *class* lainnya, ada beberapa jenis hubungan *class*, yaitu *dependency*, asosiasi, agregasi, komposisi dan generalisasi. [10]

a. *Dependency*

Dependency merupakan hubungan terlemah antar *class*. *Dependency* bermakna satu *class* menggunakan atau memiliki pengetahuan terhadap *class* lain, namun hubungannya hanya sementara dan tidak ada batas waktu yang jelas.

b. Asosiasi

Asosiasi mempunyai hubungan yang lebih kuat dari hubungan *dependency*, dimana suatu *class* tetap berhubungan dengan *class* lain seterusnya. Hubungan asosiasi dibagi menjadi dua jenis yaitu *directional* dan *bidirectional*.

c. Agregasi

Agregasi merupakan bentuk hubungan yang mengimplikasikan kepemilikan suatu *class*. Agregasi juga merupakan bentuk yang lebih kuat dari asosiasi. Hubungan agregasi dinyatakan dengan simbol *diamond* pada pemilik kelas dan garis utuh berpanah kekelas yang dimiliki.

d. Komposisi

Komposisi merupakan bentuk hubungan antar *class* yang paling kuat. Komposisi digunakan untuk mengambil seluruh bagian dari *class* yang berhubungan. Aturan dari komposisi yaitu hanya boleh ada satu komposisi dalam satu waktu.

e. Generalisasi

Generalisasi merupakan bentuk hubungan antar *class*, dari *class* yang umum dengan *class* yang lebih khusus. Contoh *class* hewan memiliki hubungan generalisasi dengan *class* kucing, karena kucing merupakan *class* khusus dari *class* hewan yang lebih umum.

2.2.4.3 Diagram *Use Case*

Diagram *use case* digunakan untuk mendeskripsikan kejadian-kejadian apa saja yang dapat dilakukan oleh *user*/aktor dan fungsionalitas-fungsionalitas apa saja yang diharapkan dari sistem yang akan dibangun, tanpa mendeskripsikan bagaimana sistem menyelesaikannya[9]. Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase requirements, model *use case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari *input user* dan respon-respon sistem. [10]

Setiap *use case* menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model *use case* secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap *use case*.

2.2.4.4 Diagram *Sequence*

Diagram *sequence* menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang menekankan pada pengiriman pesan dalam suatu waktu tertentu[9]. *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. [10]

2.2.4.5 Diagram *Collaboration*

Diagram *collaboration* menunjukan informasi yang sama seperti dalam *sequence* diagram. Perbedaannya adalah dari cara mengelompokan objek *sequence* berdasarkan urutan nomor dari pesan[9].

2.2.4.6 Diagram *statechart*

Diagram *statechart* menggambarkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. [10]

2.2.4.7 Diagram *Activity*

Diagram *activity* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir[9]. *Activity* diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.. [10]

2.2.5 Visual C#.net

Microsoft C# (disebut C sharp) adalah sebuah bahasa pemrograman yang didesain untuk membangun jangkauan aplikasi enterprise yang berjalan di atas framework .NET. Sebuah evolusi Microsoft C dan Microsoft C++, C# sederhana,

modern, aman dan Object Oriented. C# dikenal sebagai visual C# dalam Visual Studio .Net. Dukungan untuk Visual C# termasuk proyek template, desainer, halaman poperti, kode, model objek dan fitur lain dari lingkungan pengembangan. Library untuk pemrograman visual c# adalah .NET Framework.

2.2.5.1 Sejarah C#.net

Pada akhir dekade 1990-an, Microsoft membuat program Microsoft Visual J++ sebagai sebuah langkah percobaan untuk menggunakan Java di dalam sistem operasi Windows untuk meningkatkan antarmuka dari Microsoft Component Object Model (COM). Akan tetapi, akibat masalah dengan pemegang hak cipta bahasa pemrograman Java, Sun Microsystems, Microsoft pun menghentikan pengembangan J++, dan beralih untuk membuat pengganti J++, kompilernya dan mesin virtualnya sendiri dengan menggunakan sebuah bahasa pemrograman yang bersifat general-purpose. Untuk menangani proyek ini, Microsoft merekrut Anders Helsberg, yang merupakan mantan karyawan Borland yang membuat bahasa Turbo Pascal, dan Borland Delphi, yang juga mendesain Windows Foundation Classes (WFC) yang digunakan di dalam J++. Sebagai hasil dari usaha tersebut, C# pun pertama kali diperkenalkan pada bulan Juli 2000 sebagai sebuah bahasa pemrograman modern berorientasi objek yang menjadi sebuah bahasa pemrograman utama di dalam pengembangan di dalam platform Microsoft .NET Framework.

Pengalaman Helsberg sebelumnya dalam pendesain bahasa pemrograman seperti Visual J++, Delphi, Turbo Pascal) dengan mudah dilihat dalam sintaksis bahasa C#, begitu pula halnya pada inti Common Language Runtime (CLR). Dari kutipan atas interview dan makalah-makalah teknisnya ia menyebutkan kelemahan kelemahan yang terdapat pada bahasa pemrograman yang umum digunakan saat ini, misalnya C++, Java, Delphi, ataupun Smalltalk. Kelemahan-kelemahan yang dikemukakannya itu yang menjadi basis CLR sebagai bentukan baru yang menutupi kelemahan-kelemahan tersebut, dan pada akhirnya mempengaruhi desain pada bahasa C# itu sendiri. Ada kritik yang menyatakan C# sebagai bahasa yang berbagi akar dari bahasa-bahasa pemrograman lain. [1] Fitur-

fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti garbage collection, reflection, akar kelas (root class), dan juga penyederhanaan terhadap pewarisan jamak (multiple inheritance). Fitur-fitur tersebut di dalam C# kini telah diaplikasikan terhadap iterasi, properti, kejadian (event), metadata, dan konversi antara tipe-tipe sederhana dan juga objek. C# didisain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan Rapid Application Development yang 'tanpa batas' (dibandingkan dengan RAD yang 'terbatas' sepertiyang terdapat pada Delphi dan Visual Basic).

Agar mampu mempromosikan penggunaan besar-besaran dari bahasa C#, Microsoft, dengan dukungan dari Intel Corporation dan Hewlett-Packard, mencoba mengajukan standardisasi terhadap bahasa C#. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh European Computer Manufacturer Association (ECMA), dengan nomor standar ECMA-334. PadaDesember 2002, standar kedua pun diadopsi oleh ECMA, dan tiga bulankemudian diterima oleh International Organizaion for Standardization (ISO), dengan nomor standar ISO/IEC 23270:2006.[1]

2.2.5.2 Pengertian C#.net

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.[1]