

BAB II KAJIAN PUSTAKA

2.1 Teori Dasar

Menurut (Anwari & java creatifity, 2010) *internet download accelerator* merupakan sebuah *download manager* namun pada praktiknya *software* ini akan mengedepankan fungsi kecepatan dengan menambahkan sebuah *accelerator*. *Download manager* adalah aplikasi yang membantu untuk mempermudah proses men-*download*, diantaranya memudahkan untuk mengambil data dari *internet* yang berupa *file* PDF, musik, video dan yang lainnya.

2.1.1 Pengertian Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem. Menurut (Binanto, 2010) *Design* (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan, dan kebutuhan material/bahan untuk program. terdapat 2 jenis perancangan sistem yang akan di implemntasikan pada *download manager* yang pertama adalah perancangan proses, diamana menjelaskan sebuah proses bekerjanya sebuah sistem untuk melakukan pengolahan data input menjadi data

output menggunakan fungsi-fungsi yang sudah direncanakan. perancangan sistem berikutnya adalah perancangan interface, yaitu bagian dari aplikasi yang bisa dipakai oleh end user yang bisa dilihat pada layar monitor apabila program dijalankan

2.1.2 Pengertian Aplikasi

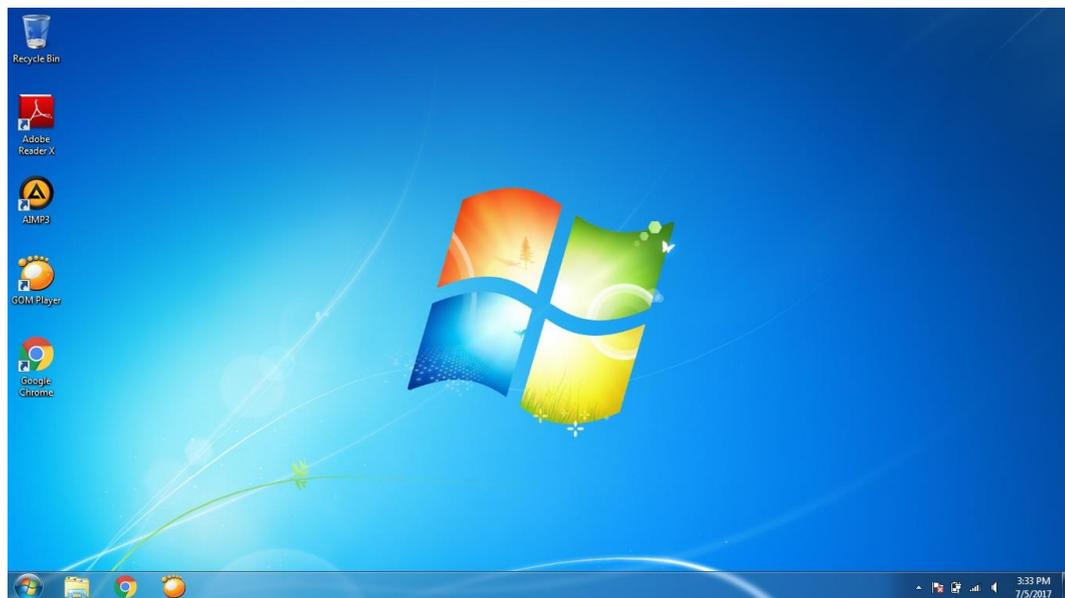
Menurut (Hendrayudi, 2009) aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Aplikasi juga dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem operasi tertentu yang berguna untuk membantu mempermudah pekerjaan.

Pengertian tentang aplikasi dapat disimpulkan bahwa aplikasi adalah sebuah program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus. Klasifikasi aplikasi dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.1.3 Pengertian *Desktop*

Menurut (Hasyim, 2009) *desktop* merupakan sebuah jendela atau gambar yang tampil pertama kali pada saat komputer atau laptop dihidupkan dan masuk pada sistem operasi berbasis *windows* XP atau Vista. Didalam tampilan *desktop*, terdapat gambar kecil diatas wallpaper yang diikuti dengan identitasnya yang disebut *icon* program. *icon* program pada *desktop* berfungsi untuk mengaktifkan suatu program.

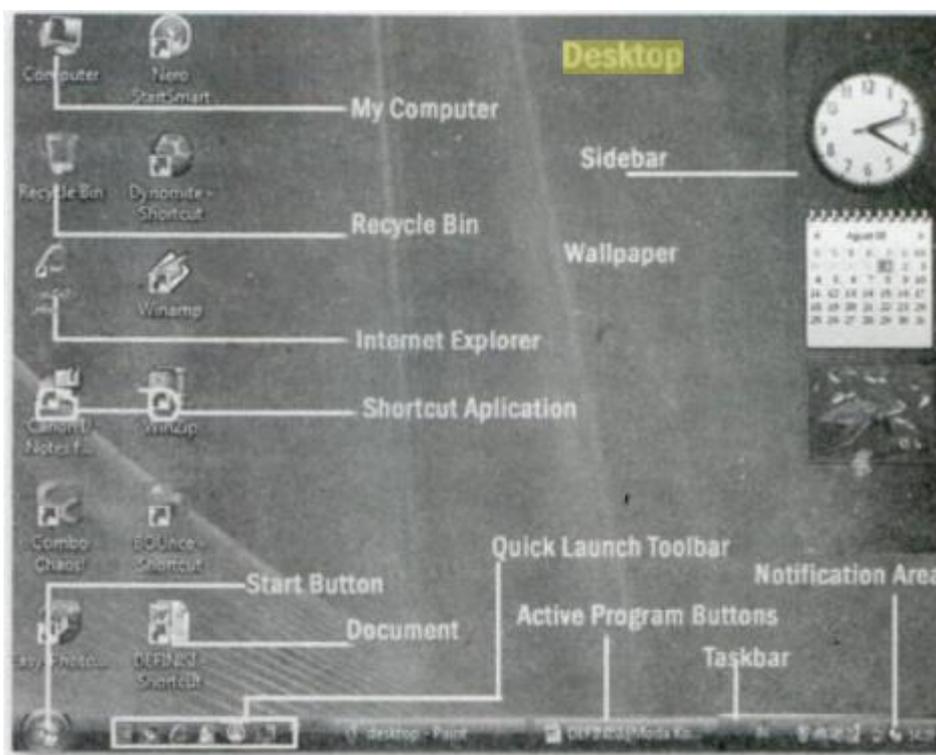


Gambar 2. 1 Tampilan *Desktop*
Sumber: Olahan Data Peneliti (2017)

Pengertian lainnya tentang *desktop* adalah menurut (Talib, 2010) *desktop* adalah tampilan layar *windows*. Umumnya *desktop* dihias dengan gambar atau foto. Pada *desktop* terdapat *icon shortcut* untuk menjalankan program. Dari pengertian *desktop* diatas dapat disimpulkan bahwa computer *desktop* adalah computer yang biasanya berada diatas meja dan ukuran yang relative besar dan memiliki input,

pemrosesan dan output secara terpisah dan dihubungkan melalui kabel maupun tanpa kabel (*wireless*). Pengertian tentang *desktop* sendiri dapat disimpulkan bahwa *desktop* adalah halaman paling depan saat membuka *windows*, yaitu halaman pertama yang ditampilkan *windows* setelah selesai *loading* atau *booting*.

Ada banyak bagian-bagian pada *desktop* seperti *recycle bin*, *my computer*, *wallpapaer*, *shortcut application*, dan lain-lain.



Gambar 2. 2 Bagian-bagian *Desktop*

Sumber: (Hasyim, 2009)

Bagian-bagian *desktop* dapat dilihat pada gambar diatas. Menurut (Hasyim, 2009) bagian bagian *desktop* adalah sebagai berikut:

1. *Recycle Bin*

Icon ini berfungsi seperti tong sampah. setiap kali terjadi penghapusan *file*, baik yang sengaja atau tidak sengaja maka *file-file* tersebut belum benar-benar dihapus, tetapi masih tersimpan di dalam recycle bin. selama *file* masih tersimpan pada *recycle bin* maka dapat dikembalikan dengan mengklik *icon* recycle bin, lalu klik menu restore item. *file-file* yang terhapus akan kembali pada posisi awal sebelum terhapus.

2. *My Computer*

Icon ini berfungsi menampilkan drive atau *directory* yang ada di dalam komputer serta pengolahan *file* dan *folder* pada *windows*, seperti perintah copy, *format*, delete, dan cut.

3. *My Network Places*

Icon ini berfungsi menampilkan daftar nama komputer yang terhubung dalam jaringan atau *network*

4. *My Document*

Merupakan suatu folder standar pada *windows* tempaan penyimpanan semua data secara otomatis jika pengguna komputer menspesifikkan tempatnya pada saat penyimpanan data. dengan mengklik *icon* ini, data dapat langsung diakses dengan cepat.

5. *Document*

Icon file sebuah lembar kerja atau *file* dari salah satu program aplikasi pada *desktop windows* yang dibuat untuk mempercepat pembukaan lembar kerja

file tersebut tanpa harus membuka program atau aplikasi yang terkait dengan *file* kerja tersebut

6. *Internet Explorer*

Icon ini berfungsi mengaktifkan *software internet explorer*. *internet explorer* adalah *software* yang berfungsi untuk menjelajahi dunia maya(*internet*). jika komputer tidak terhubung ke *internet* maka *software* ini tidak berfungsi sebagaimana mestinya.

7. *Shortcut Application*

Sebuah *icon* program atau aplikasi yang dibuat saat meng-install sebuah program atau aplikasi yang berguna mempercepat pengguna komputer untuk membuka aplikasi tersebut dengan *double click* tanpa harus melalui tombol menu *start* dan program.

8. *Quick Launch Toolbar*

sebuah *toolbar* yang berfungsi menampilkan beberapa *icon* aplikasi atau program yang berlangsung bila dijalankan dengan sekali klik gambar *icon* pada *toolbar* tersebut.

9. *Active Program Buttons*

Icon minimize program atau aplikasi pada *toolbar* yang menjelaskan bahwa program tersebut sedang aktif digunakan atau menampilkan program yang sedang diaktifkan oleh pengguna komputer.

10. *Taskbar*

Sebuah batang panjang pada paling bawah *desktop* yang dapat dipindahkan ke bagian atas, samping kanan, dan kiri *desktop* sesuai keinginan . pada

taskbar ini terdapat program yang dapat di-*minimize*-kan, *icon*-*icon* quick launch, serta *icon* pengaturan *volume* dan jam yang aktif.

11. *Notification Area*

lokasi taskbar palingkanan yang berisi beberapa *icon* yang ikut aktif saat komputer dihidupkan, seperti *icon* jam, volume, dan antivirus, *icon* ini dapat langsung di klik kanan atau kiri untuk mengatur setting atau konfigurasi dari setiap *icon* program yang ada pada *taskbar* ini.

12. *Sidebar*

Mini bar di sebelah kanan *desktop* dan berbentuk memanjang ke bawah yang hanya ada pada *windows vista*. *sidebar* ini berguna menampilkan jam, tanggal, gambar *slideshow*, catatan harian dan beberapa aplikasi lain yang diinginkan.

13. *Start Button*

Tombol yang di klik menampilkan menu *all program* (berisi semua aplikasi atau program yang telah ter-*install* di dalam komputer), *control panel*, *document*, *search*, dan lain lain.

2.2 Teori Khusus

Download adalah proses transmisi sebuah *file* dari sebuah sistem komputer ke sistem komputer yang lainnya. Menurut (Priyatno, 2009) *download* adalah istilah yang digunakan saat proses men-*transfer* data dari komputer *server* ke komputer lokal. Dapat disimpulkan bahwa *download* adalah sebuah proses dalam pengambilan

file-file tertentu yang terdapat di *internet* baik melalui web server, FTP server, mail server, server ataupun sistem lain yang identik. *file-file* yang biasanya di *download*(unduh) dari *internet* dapat berupa *file video*, Mp3, 3gp, *document*, aplikasi, *software*. Untuk mempermudah proses *download* bisa menggunakan *internet download accelerator* atau tool yang bisa mempercepat proses *download* dari pada dengan cara manual ataupun melalui browser.

2.2.1 Visual C#

Menurut Jud (2016:1) Visual C# adalah bahasa pemrograman berbasis objek yang dapat digunakan untuk mengembangkan berbagai aplikasi. Visual C# merupakan bahasa pemrograman yang dirancang untuk berkerja dilingkungan .NET Framework. Menurut (Kurniawan, 2010) C# adalah pengembangan dari bahasa pemrograman sebelumnya, yaitu C/C++, java, dan visual basic. Visual C# sendiri merupakan hasil evolusi dari Microsoft C dan Microsoft C++. Seperti semua Bahasa lainnya, C# memiliki aturan syntax dan kode-kode yang bisa digunakan untuk membuat aplikasi. Bahasa pemrograman C# 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. Tujuan Bahasa C# adalah untuk menyediakan tool untuk pengembangan program yang simple, aman berbasis objek, interner-cenric, dan mempunyai *performa* yang handal.

1. Kelebihan Visual C#:

- a) Merupakan induk dari bahasa pemrograman perl, php, python, visual basic, gmbas, java
- b) Compiler bahasa C# terdapat di semua *platform*
- c) Untuk pengembangan visual dijejali dengan *platform* yang sangat banyak seperti OWL, MFC, Cocoa, QT, GTK, dll
- d) Visual C# merupakan pemrograman berorientasi objek

2. Kekurangan Visual C#:

- a) Bahasa ini cukup sulit untuk dipelajari dan dipahami
- b) Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai
- c) Bagi pemula pada umumnya akan kesulitan menggunakan pointer

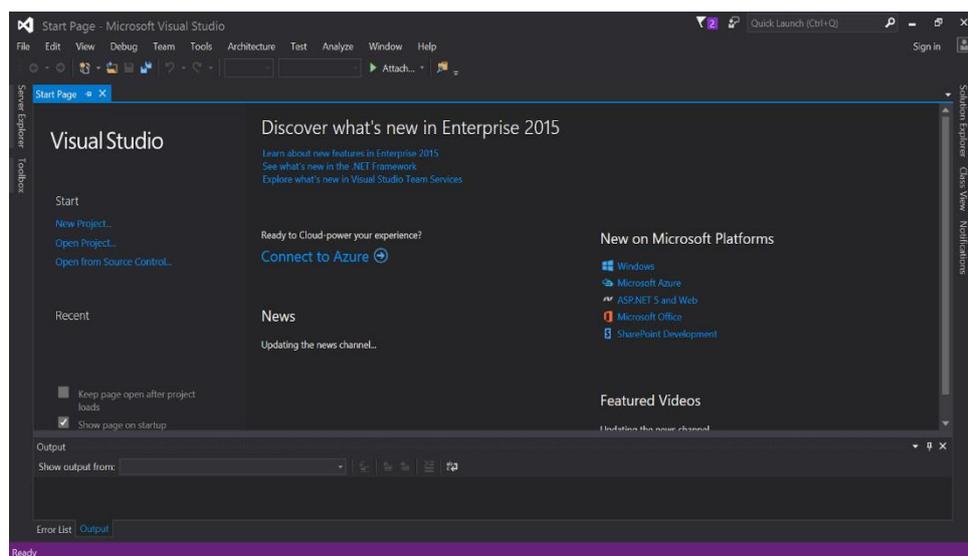
2.3 Tools

Adapun *tools* atau alat-alat yang digunakan dalam pembuatan skripsi analisis dan perancangan aplikasi *download* manager ini adalah sebagai berikut.

2.3.1 Visual Studio 2015 IDE

Visual Studio dapat digunakan untuk mengembangkan aplikasi *mobile*, *web*, *desktop* dan *cloud*. . 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*. namun Visual Studio hanya bisa digunakan pada sistem operasi *windows*. 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*).



Gambar 2. 3 Tampilan Visual Studio 2015
Sumber: Olahan Data Peneliti (2017)

2.3.2 *Unified Modeling Language (UML)*

Menurut (Mulyani, 2017) UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. UML muncul karena adanya kebutuhan

pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

2.3.2.1 Use Case Diagram

Use case diagram, yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor (Mulyani, 2017). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Menurut (Rosa & Shalahuddin, 2013) syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah symbol-simbol yang ada pada diagram *use case*:

Tabel 2. 1 Simbol-Simbol Pada Diagram Use Case

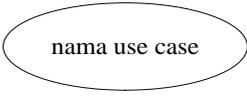
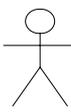
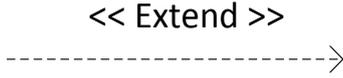
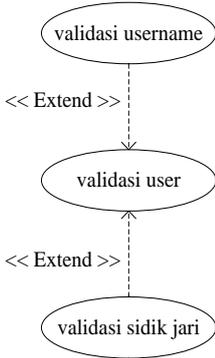
Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

Table 2.1 Lanjutan

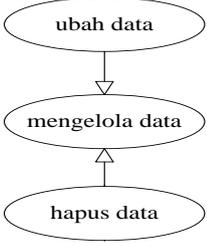
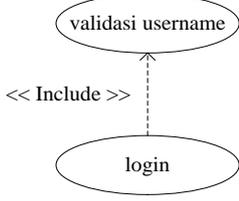
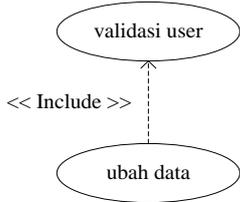
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasainya (umum).</p>
<p>Menggunakan / <i>include / uses</i></p> <p><< Include >></p>  <p><< Uses >></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat diljalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <ol style="list-style-type: none"> 2. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan,

Table 2.1 Lanjutan

	<p>3. sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut:</p>  <pre> graph BT UC1(ubah data) -.-> << Include >> UC2(validasi user) </pre>
--	---

Sumber: (Rosa & Shalahuddin, 2013)

1.3.2.2 Activity Diagram

Menurut (Rosa & Shalahuddin, 2013) diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) 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, jadi aktivitas yang dapat dilakukan oleh sistem.

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

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

3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2. 2 Simbol-Simbol Pada Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas yang biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Sumber: (Rosa & Shalahuddin, 2013)

1.3.2.3 *Class Diagram*

Menurut (Rosa & Shalahuddin, 2013) diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar dokumentasi perancangan dan perangkat lunak sinkron. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem (*View*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

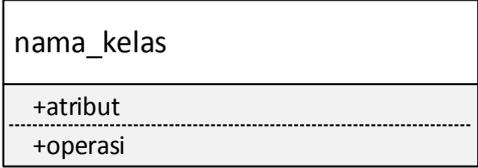
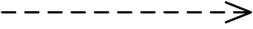
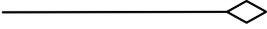
4. Kelas yang diambil dari pendefinisian data (*model*)

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

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam

hubungan-hubungan objek tersebut. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2. 3 Simbol-Simbol Pada Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep interface dalam pemrograman berorientasi objek
<p>Asosiasi berarah / <i>direct association</i></p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan / <i>dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan kelas
<p>Agresi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

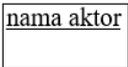
Sumber: (Rosa & Shalahuddin, 2013)

1.3.2.4 Sequence Diagram

Menurut (Rosa & Shalahuddin, 2013) diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

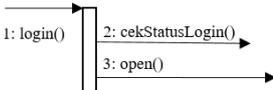
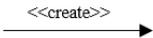
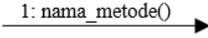
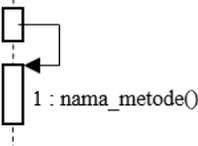
Ada banyak symbol-simbol diagram sekuen seperti symbol actor, garis hidup, objek, waktu aktif. Menurut (Rosa & Shalahuddin, 2013) simbol-simbol yang ada pada diagram sekuen adalah sebagai berikut.

Tabel 2. 4 Simbol-Simbol Pada Sequence Diagram

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem <i>informasi</i> yang akan dibuat di luar sistem <i>informasi</i> yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>

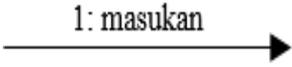
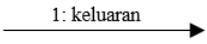
Sumber: (Rosa & Shalahuddin, 2013)

Tabel 2.4 Lanjutan

<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 10px auto;"> <p>nama objek : nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login() Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain ata dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>

Sumber: (Rosa & Shalahuddin, 2013)

Tabel 2.4 Lanjutan

<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/<i>informasi</i> ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

Sumber: (Rosa & Shalahuddin, 2013)

2.4 Penelitian Terdahulu

Dari berbagai penelitian yang telah dilakukan terdahulu yang digunakan sebagai dasar pengembangan dalam pembuatan penelitian ini antara lain:

1. Nama Jurnal : Jurnal Teknologi Informasi DINAMIK

Judul Jurnal : Perancangan dan Pengaplikasian Sistem Penjualan pada “*Distro Smith*” Berbasis *E-Commerce*

Nama Peneliti : Rara Sri Artati Rejeki, Agus Prasetyo Utomo, dkk.

Volume / Tahun / ISSN : Vol 16/2016/ 0854-9524

Kesimpulan : Aplikasi aplikasian Sistem Penjualan pada “*Distro Smith*”

Berbasis *E-Commerce* di rancang dengan menggunakan bahasa

pemrograman PHP *text editor Macromedia Dreamweaver, Adobe CS4* dan database MySQL. Adanya *website* pemasaran ini dapat membantu perusahaan dalam meningkatkan pelayanan dan penjualan di berbagai wilayah baik dalam kota maupun luar kota. Aplikasi Penjualan *Online Distro Smith Semarang* menyajikan laporan-laporan diantaranya laporan produk, laporan member, laporan order, dan laporan penjualan. Laporan tersebut ditujukan kepada manajer sebagai bahan pertimbangan dalam perbaikan pelayanannya.

2. Nama Jurnal : Pelita Informatika Budi Darma

Judul Jurnal : Perancangan Aplikasi Untuk Memeriksa Keaslian Data Yang Telah Di *download* Menggunakan Algoritma *Message Digest 5 (Md5)*

Nama Peneliti : Sulianto

Volume / Thn / ISSN : Vol 8 / 2014 / 2301-9425

Kesimpulan : dalam jurnal perancangan aplikasi untuk memeriksa keaslian data yang telah di *download* menggunakan algoritma *message digest 5(MD5)*, dapat disimpulkan bahwa MD5 dapat mendeteksi perubahan yang terjadi pada suatu *file* walaupun perubahan tersebut sangat kecil. Perubahan dapat terdiri dari penghapusan, penggantian atau penambahan dari suatu *file*. MD5 dapat membangkitkan kode dari segala *format file* termasuk zip, exe, iso dan *file-file* lainnya. Kecepatan pembangkitan kode MD5 dari suatu *file* tergantung dari besar ukuran *file*

tersebut. Semakin besar ukuran dari suatu *file*, maka semakin lama kode MD5 yang dihasilkan, demikian juga sebaliknya.

3. Nama Jurnal : Jurnal Nasional Pendidikan Teknik Informatika

Judul Jurnal : Perancangan MP3 *Player* dengan Visual C# 2010

Nama Peneliti : Megawaty dan Oktaviani

Volume / Thn / ISSN : Vol 4 / 2015 / 2087-2658

Kesimpulan : Dalam jurnal perancangan MP3 *Player* dengan visual C# 2010 dapat disimpulkan *Unified Modeling Language* memberikan visualisasi serta panduan yang jelas dalam proses perancangan perangkat lunak MP3 *Player*, hingga dapat terealisasi program yang dapat menjalankan *file* mp3 untuk didengarkan melalui perangkat komputer multimedia. Tombol-tombol yang terdapat pada tampilan utama program MP3 *Player* dapat berjalan dengan baik dan benar sesuai dengan perancangan yang dibuat, tombol-tombol tersebut berjalan sesuai dengan fungsi serta prosedur yang diberikan. Selain tombol-tombol terdapat animasi berupa *visualizer* yang menunjukkan bahwa musik sedang diputar untuk menambah kesan menarik pada MP3*Player*.

4. Nama Jurnal : Jurnal TIMES

Judul Jurnal : Perancangan Aplikasi *E-Library* Menggunakan Bahasa Pemrograman PHP Pada Universitas Methodist Indonesia

Nama Peneliti : Ambon Saragih, Emma Rosinta Simarmata, dkk.

Volume / Thn / ISSN : Vol 4 / 2015 / 2337-3601

Kesimpulan : Perancangan Aplikasi *E-Library* Menggunakan Bahasa Pemrograman PHP menggunakan data yang ada pada perpustakaan Universitas Methodist Indonesia seperti data buku, data transaksi peminjaman dan laporan. Proses yang dilakukan dalam perancangan aplikasi yaitu dengan membuat *Data Flow Diagram* (DFD), rancangan database, *flowchart*, dan rancangan tampilan menu.

5. Nama Jurnal : Jurnal Teknologi Informasi & Pendidikan

Judul Jurnal : Aplikasi Proyek Pnpm Mpd Kec. Siulak Kab. Kerinci

Nama Peneliti : Defiariany

Volume / Thn / ISSN : Vol 9 / 2016 / 2086 – 4981

Kesimpulan : Berdasarkan hasil penelitian dan pembahasan dari jurnal ini dapat diambil kesimpulan bahwa Aplikasi Proyek PNPM MPd Kec.Siulak Kab.Kerinci adalah aplikasi berbasis *desktop*. Database yang digunakan dalam penyimpanan data adalah MySQL dan Bahasa pemrograman C#. Aplikasi ini dapat membantu pihak Petugas PNPM dalam pengisian data, tanpa mengisikan data secara berulang. Mempermudah pencarian data proyek, tidak membutuhkan waktu lama dalam menghasilkan laporan yang diinginkan, Pengguna dan Petugas Adm dan keuangan yang mempunyai

6. Nama Jurnal : *Scalable Computing: Practice and Experience*

Judul Jurnal : *Minimization Of Download Time Variance In A Distributed Vod System*

Nama Peneliti : Anne, Vincent Dkk

Volume / Thn / ISSN : Vol 10 / 2009 / 1895-1767

Kesimpulan : analisis dengan berbagai simulasi algoritma untuk mengoptimalkan waktu varians *download* dalam arsitektur distribusi, dengan menempatkan replikasi sesuai data *server* terdistribusi. secara khusus, telah ditunjukkan bahwa putaran konvensional algoritma *round robin* secara umum tidak berjalan dengan baik, dan algoritma acak adalah pilihan yang terbaik,

7. Nama Jurnal : *International Journal Of Engineering And Computer Science*

Judul Jurnal : *Comparative study of C, Objective C, C++ programming language*

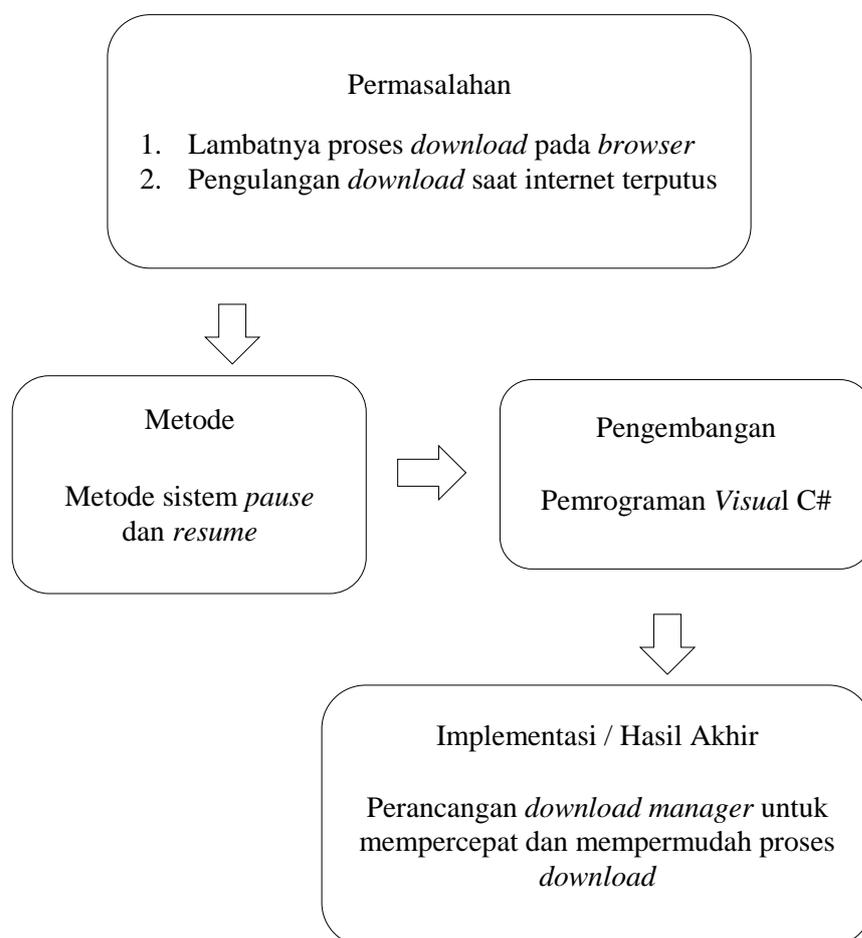
Nama Peneliti : Partibha dan Ashu

Volume / Thn / ISSN : Vol 2 / 2015 / 2319-7242

Kesimpulan : C# adalah kombinasi dari semua bahasa pemrograman lainnya dalam keseimbangan yang hampir sempurna. C# adalah bahasa yang berorientasi objek murni. Sintaks ringkas C juga ditambahkan ke dalamnya. Sintaks C# lebih mirip dengan *Java* daripada ke C++. Manajemen memori *pointer* di C# tidak menjadi masalah lagi.

2.5 Kerangka Berpikir

Menurut (Fitrianti, 2016) kerangka berfikir adalah penjelasan sementara terhadap suatu gejala yang menjadi objek permasalahan. Gambar 2.4 adalah kerangka pemikiran yang mendasari penelitian ini.



Gambar 2. 4 Kerangka pemikiran
Sumber: Data Penelitian (2017)

Lambatnya proses *download* pada *browser windows* seperti *Internet Explorer* membuat aktivitas *download* memerlukan waktu yang cukup lama dan

download akan mengulang kembali jika *internet* terputus, karena permasalahan tersebut peneliti ingin membuat sebuah aplikasi *download manager* dengan menggunakan pendekatan metode sistem *pause* dan *resume*. Pengembangan aplikasi *download manager* tersebut menggunakan aplikasi pengembang yaitu *Visual Studio 2015* Perancangan *download manager* ini bertujuan untuk mempermudah proses *download* dan mempercepat proses *download* dari pada proses *download* di *browser*.