BAB II

TINJAUAN PUSTAKA

A. PENELUSURAN REFRENSI

Dari beberapa jurnal, penulis memaparkan dua jurnal yang akan menjadi sumber referensi yang telah dibangun sebelumnya, yang pertama penulis menelusuri jurnal aplikasi yang telah dibangun oleh Didik Setiawan,Yhoni Agus Setya Mahendra (2014) "Perancangan Sistem Informasi Penduduk Pada Kantor Desa Kebonsari" di dalam aplikasi tersebut terdapat beberapa yang fitur disajikan, diantaranya adalah menu kependudukan. Dari fitur yang disediakan, admin dapat memilih dan menjalankan fitur yang diinginkan. yang kedua penulis menelusuri aplikasi yang telah dibangun oleh Hermawan Susanto (2013)"Aplikasi Pendataan Arsip Dan Administrasi Di Kelurahan Kebon Jeruk Berbasis java".Di dalam aplikasi yang dibangun terdapat fitur kependudukan yang hampir sama dengan jurnal sebelumnya.

Dari kedua jurnal tersebut terdapat fitur yang belum ada,yaitu pengklasifikasian kesejahteraan penduduk.Oleh karena itu penulis ingin membangun **"**Perancangan Aplikasi Pendataan Arsip dan Administrasi Di Kelurahan Desa Joho Kabupaten Madiun Berbasis Java Neatbeans**".**

B. PENGERTIAN ARSIP

Arsip adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh lembaga negara, pemerintahan daerah, lembaga pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam

pelaksanaan kehidupan bermasyarakat, berbangsa, dan bernegara.(UU No. 43 Tahun 2009 tentang Kearsipan).

C. ADMINISTRASI KEPENDUDUKAN

Penataan administrasi direkomendasikan untuk penyelenggaraan registrasi penduduk termasuk pemberian Nomor Induk Kependudukan (NIK). Dalam pelaksanaan sistem ini, semua penduduk baik Warga Negara Indonesia (WNI) maupun warga Negara Asing (WNA) yang mengalami kejadian vital atau perubahan status kependudukannya harus mendaftarkan diri atau mencatatkan perubahan statutersebut kepada para petugas yang ditunjuk oleh negara. Dengan adanya sistem ini, pemerintah akan memperoleh kemudahan dalam mengatur bentuk-bentuk pelayanan publik lainnya misalnya dibidang pendidikan, kesehatan dan sebagainya.Dalam peraturan pemerintah pada Undang-undang No. 23 Tahun 2006 tentang Administrasi Kependudukan yang dimaksud dengan Administrasi kependudukan adalah ” Administrasi Kependudukan adalah rangkaian kegiatan penataan dan penerbitan dokumen dan data kependudukan melalui program pendaftaran penduduk, pencatatan sipil, pengelolaan informasi administrasi kependudukan serta pendayagunaan hasilnya untuk pelayanan publik dan pembangunan sektor lain ”. (UU No. 23 Tahun 2006 : 4).

D. MYSQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web *(wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional

MySQL pada awalnya diciptakan pada tahun 1979, oleh *Michael "Monty" Widenius,* seorang programmer komputer asal Swedia. *Monty* mengembangkan sebuah sistem database sederhana yang dinamakan *UNIREG* yang menggunakan koneksi low-level ISAM database *engine* dengan *indexing*. Pada saat itu Monty bekerja pada perusahaan bernama *TcX di Swedia*.TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggagap tidak cocok untuk database yang dinamis seperti web.TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di ‘jual’ dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa

“support” untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

E. JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh *James Gosling* saat masih bergabung d*i Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

Java terlahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh *Patrick Naughton, Mike Sheridan*, dan *James Gosling*, beserta sembilan pemrogram lainnya dari *Sun Microsystems*. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh *Joe Palrang*.Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program

Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.Mereka menjadikan perambah (*browser*) *Mosaic* sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel *Sheraton Palace*. Tiga dari pimpinan utama proyek, *Eric Schmidt* dan *George Paolini* dari *Sun Microsystems* bersama *Marc Andreessen*, membentuk *Netscape*.Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau

Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan di atas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut,OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek),Perpustakaan Kelas Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi,Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.Pengumpulan sampah

otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

F. METODE PENDEKATAN WATERFALL

Model pendekatan *waterfall* menawarkan cara pembuatan perangkat lunak secara lebih nyata. Langkah – langkah yang penting dalam teknik pendekatan *waterfall* ini adalah

- Penentuan dan analisis spesifikasi

  Jasa, kendala dan tujuan dihasilkan dari konsultasi dengan pengguna sistem. Selanjutnya seluruh konsultasi itu dibuat dalam bentuk yang mudah untuk dimengerti oleh staf pengembang dan user.

- Desain sistem dan perangkat lunak

  Dalam proses *desain* sistem membagi kebutuhan – kebutuhan menjadi sistem perangkat lunak dan perangkat keras, dalam proses ini menghasilkan sebuah arsitektur sistem keseluruhan. Fungsi dari desain perangkat lunak adalah menghasilkan sistem perangkat lunak dalam dalam bentuk yang yang mungkin ditransformasi ke dalam satu atau lebih program yang dapat dijalankan.

- Implementasi dan uji coba unit

  Dalam tahapan ini desain perangkat lunak disadari sebagai sebuah program lengkap atau unit program. Uji coba unit termasuk dalam pengujian setiap unit sesuai spesifikasi.

- Integrasi dan dan uji coba sistem

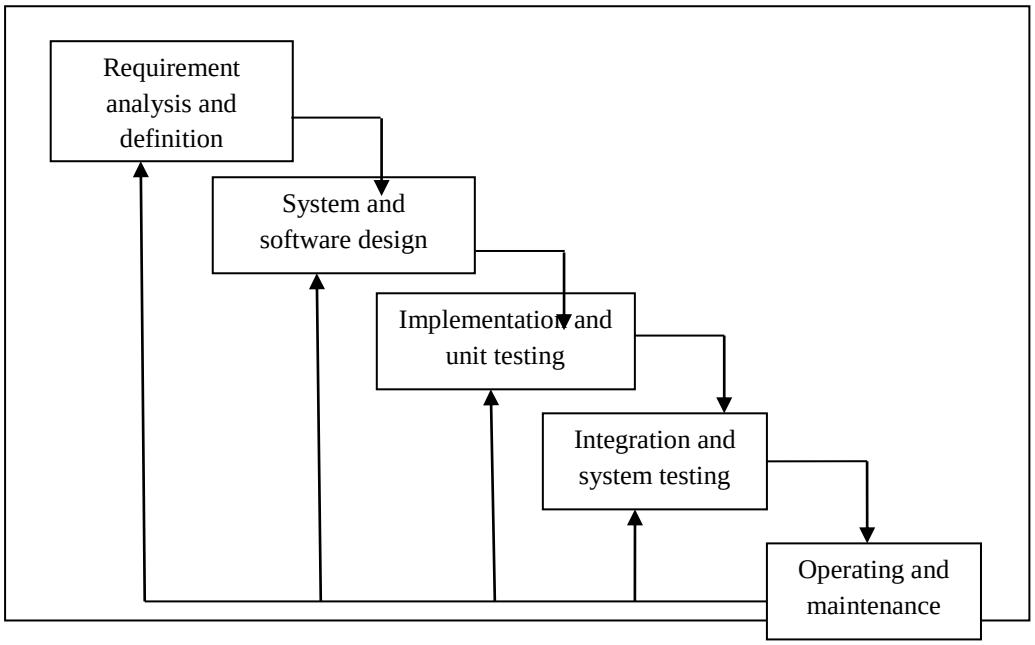
  Unit program akan diintegrasikan dan diuji menjadi sistem yang lengkap bertujuan untuk meyakinkan bahwa persyaratan perangkat lunak telah

terpenuhi. Apabila telah melalui uji coba maka akan di sampaikan ke *customer*.

- Operasi dan pemeliharaan

  Pada tahap ini sistem akan dipasang dan digunakan.pemeliharaan akan dilakukan dengan cara pembetulan kesalahan yang sebelumnya belum ditemukan. Perbaikan implementasi suatu unit sistem dan peningkatan mutu jasa sistem yang baru ditemukan.

Model *waterfall* mencerminkan kepraktisan *engineering*. Masalah pendekatan waterfall adalah ketidak luwesan pembagian projek ke dalam langkah yang nyata atau jelas, sistem yang disampaikan terkadang tidak memenuhi keinginan customer. Langkah-langkah pentingdalam model pendekatan *waterfall* tersebut adalah



Sumber : Al-bahra bin Ladjamuddin, (2006:18)

Gambar 2.1 Siklus hidup (life cycle) dengan model-model waterfall

Aktifitas dan dokumentasi pada model waterfall :

Aktivitas	Dokumen keluaran
Analisa kebutuhan	Study kelayakan Garis besar kebutuhan
Definisi kebutuhan	Spesifikasikebutuhan
Spesifikasi system	Spesifikasi fungsional Spesifikasi uji penerimaan Draft / usulan manual pengguna

Sumber : Al-bahra bin Ladjamuddin, (2006:18

G. *Unified Modeling Language* (UML)

"UML *(Univied Modeling Language)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma berorientasi objek" (Adi Nugroho,2010:6)

Sehingga dapat disimpulkan, UML *(Univied Modeling Language)* adalah pemodelan objek yang digunakan untuk menyajikan suatu sistem yang berorientasi pada objek.

UML merupakan sintak umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dipahami selama fase analisis dan desain.UML biasanya disajikan dalam bentuk diagram/gambar yang meliputi *class* beserta atribut dan operasinya.

1. FLOWCHART

Flowchart adalah sebuah bagan–bagan yang mempunyai arus yang menggambarkan langkah–langkah penyelesaian suatu masalah, cara ini merupakan penyajian dari suatu algoritma.

a. Macam – macam *Flowchart* dalam proses dengan computer

1) System flowchart

Bagan ini menampilkan urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpana dalam proses pengolahan data.

2) Program *flowchart*

Yaitu sebuah bagan yang menampilkan urutan suatu intruksi yang digambarkan dengan symbol tertentu bertujuan untuk memecahkan masalah yang terjadi dalam sebuah program.

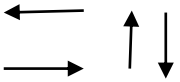
b. Simbol – symbol flowchart

Di dalam sebuah *flowchart* terdapat symbol, symbol ilah yang dipakai sebagai alat untuk membantu menggambarkan proses di dalam program. *Symbol – symbol flowchart* dibagi menjadi 3 kelompok :

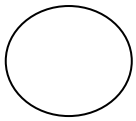
1) *Flow direction symbols* (symbol penghubung / alur)

Yaitu simbol yang dipergunakan untuk menunjukkan alur sistem. Symbol ini juga disebut *connecting line*.

Tabel 2.2 simbol-simbol *flow direction*.

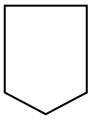
Simbol	**Keterangan**
	**Symbol arus / flow** Digunakan untuk menyatakan arus suatu proses
	**Symbol communication link**

Untuk menyatakan bahwa adanya transisi suatu data / informasi dari satu lokasi lainnya.



Symbol connector

Untuk menyatakan sambungan sari satu proses ke proses lain dalam halaman / lembar yang sama.



Symbol offline connector

Untuk menyatakan sambungan dari suatu proses lain dalam halaman / lembar yang berbeda.

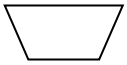
Al-bahra bin Ladjamuddin, (2006:18)

2) *Processing symbols* (symbol proses)

Yaitu symbol yang menunjukkan proses yang terjadi, menunjukkan jenis operasi pengolahan dalam prosedur.

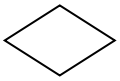
Table 2.3 symbol-simbol *Processing*

Simbol	Keterangan
	***Symbol offline connector*** Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman / lembar yang berbeda.



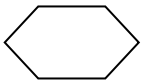
***Symbol* manual**

Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh computer (manual)



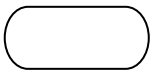
***Symbol decision* / logika**

Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban , ya / tidak



***Symbol predefined* proses**

Untuk menyatakan penyediaan tempat penyimpan suatu pengolahan untuk member harga awal.

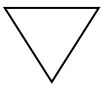


Symbol terminal

Untuk menyatakan pemulaan atau akhir suatu program

Symbol keying operating

Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard*



Symbol off-line storage

Untuk menunjukkan bahwa dalam data symbol ini akan disimpan ke suatu media tertentu

Symbol maual input

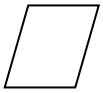
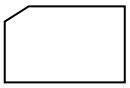
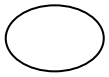
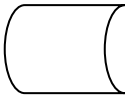
Untuk memasukkan data secara manual dengan menggunakan *online keyboard*

Al-bahra bin Ladjamuddin, (2006:18)

3) *Input – output symbol* (symbol *input- output*)

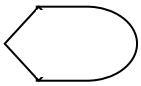
Yaitu symbol yang menunjukkan jenis peralatan yang digunakan sebagai media input atau *output*.

Tabel 2.4 Symbol –symbol *Input – output*

Simbol	**Keterangan**
	***Symbol input – output*** Untuk menyatakan proses input dan *output* tanpa tergantung dengan jenis peralatannya.
	***Symbol punched card*** Untuk menyatakan input berasal dari kartu atau *output* ditulis ke kartu.
	***Symbol magnetic-tape unit*** Untuk menyatakan input berasal dari pita magnetic atau *output* disimpan ke pita magnetic
	***Symbol disk storage*** Untuk menyatakan input berasal dari *disk* atau *output* disimpan ke disk.

Symbol document

Untuk mencetak laporan ke printer

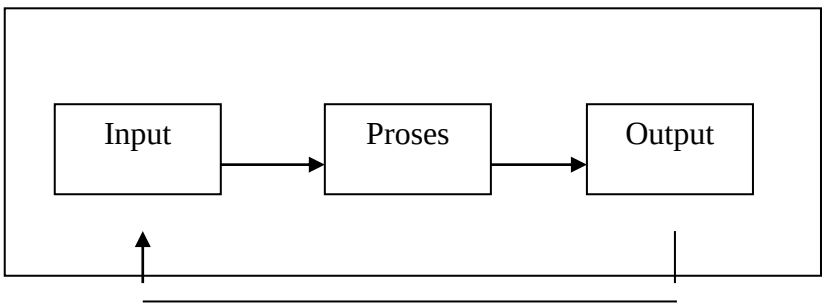


Symbol display

Untuk menyatakan peralatan output yang digunakan berupa layar (video, computer).

Al-bahra bin Ladjamuddin, (2006:18)

Dalam pembuatan *flowchart* tidak ada rumus atau kaidah baku yang bersifat mutlak, hal ini karena *flowchart* merupakan gambaran hasil pemikiran dalam menganalisa suatu masalah dengan computer, sehingga *flowchart* yang dihasilkan dapat berfariasi antara satu program dengan pemogram lainnya. Namun secara umum, setiap pengolahan *flowchart* terdiri dari 3 bagian utama.



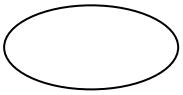
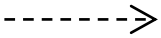
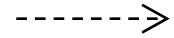
Al-bahra bin Ladjamuddin, (2006:18)

Gambar 2.2 pengolahan utama dalam *flowchat*

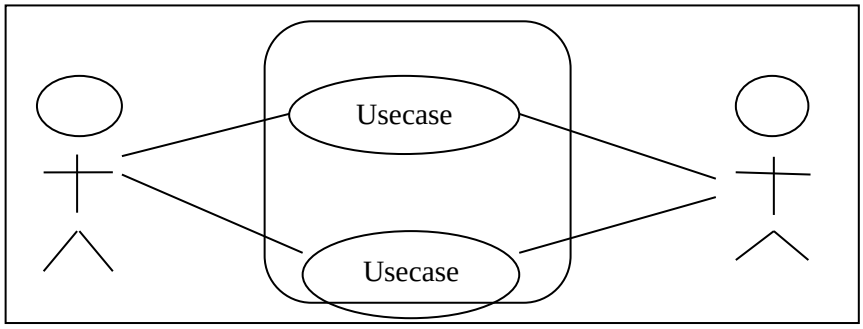
2. Use Case

Use case menurut *Satzinger et al*., (2010, p.242) "merupakan suatu aktivitas yang dilakukan oleh sistem, biasanya merupakan sebuah respon untuk permintaan dari pengguna sistem". *Satzinger et al*., (2010, p.243) "menjelaskan bahwa aktor tidak selalu sama dengan sumber dari peristiwa di *event table* karena *actor di use case* merupakan orang yang berinteraksi dengan sistem yang mana sistem harus meresponnya".

Tabel 2.5 relasi- relasi dalam *Usecase*

relasi	Fungsi	Notasi
Aktor	Digunakan untuk menggambarkan seseorang atau sesuatu.	
Use case	Digunakan untuk pemberian nama dalam kegunaan sistem yang akan dilakukan.	
Asosiasi (association)	Lintasan komunikasi antara actor dengan usecase	
Extend	Penambahan perilaku ke suatu use case dasar	
Generalisasi use case	Menggambarkan hubungan antara use case yang bersifat umum ke yang lebih spesifik	
Include	Penambahan perilaku ke suatu use case dasar secara eksplisit mendeskripsikan penambahan tersebut	

Verdi Yasin, (2012:270).



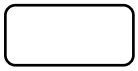
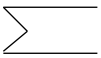
Gambar 2.3 *usecase* umum

3. Activity Diagram

Activity diagram menurut Martin Fowler (2005 : 163) adalah "teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja". sedangkan Menurut *Satzinger et al* (2010:141), "Activity diagram merupakan sebuah tipe dari diagram workflow yang menggambarkan tentang aktivitas dari pengguna ketika melakukan setiap kegiatan dan aliran sekuensial". Dalam beberapahal, *activity diagram* memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel. Node* pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*.

Tabel 2.6 simbol *activity diagram*

Simbol	Keterangan
	Titik Awal
	Titik Akhir

	Activity
	Pilihan untuk pengambilan keputusan
	Fork; untuk menunjukkan kegiatan yang dilakukan secara paralel
	Rake;menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Penerimaan
	Aliran Akhir (*Flow Final*)

Verdi Yasin, (2012:271).

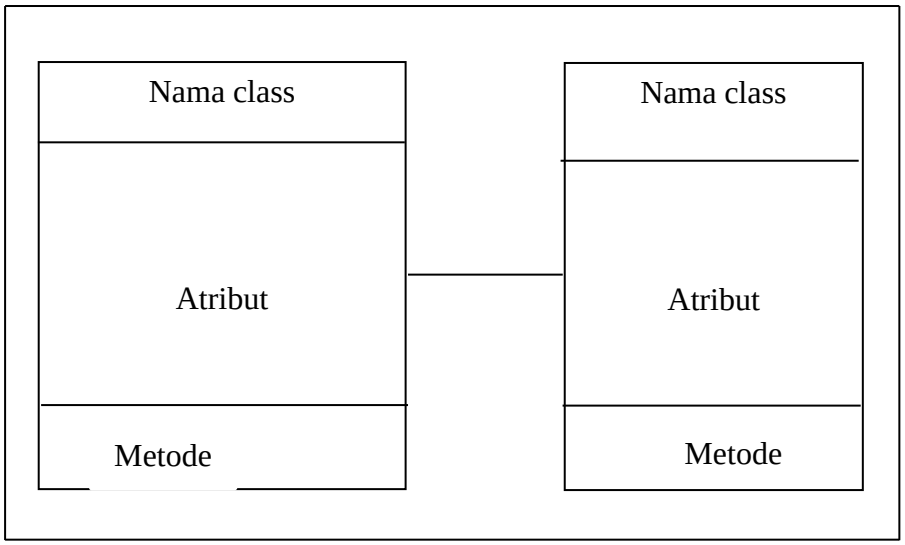
4. Diagram Class

Class diagram menurut Munawar (2005 : 28) "merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*)". *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

Objek adalah niali tertentu dari setiap *attribute* kelas *entity*.

Class memiliki tiga area pokok :

a. Nama (dan *stereotype*)
b. Atribut
c. Metode



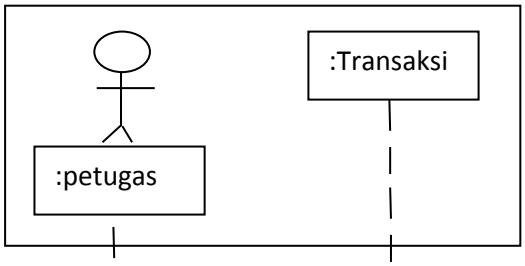
Verdi Yasin, (2012:274)

Gambar 2.4 *Class Diagram*

5.*Sequence Diagram*

Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram menjelaskan interaksi objek yang telah disusun berdasarkan urutan waktu. *Sequence* diagram erat kaitannya dengan *usecase* diagram dimana 1 *usecase* akan menjadi 1 *sequence* diagram. Dalam *sequence* terdapar 2 simbol yaitu :

a. *Actor*, untuk menggambarkan pengguna sistem
b. *lifeline* , untuk menggambarkan kelas dan objek.



Prabowo Pujo Widodo& Herlawati; (2011;15)

Gambar 2.5 *actor* dan *Life Line*