

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Perusahaan**

Berikut adalah tinjauan perusahaan mulai dari profil , visi-misi dan logo perusahaan.

##### **2.1.1 Profil Perusahaan dan Logo Perusahaan**

Datadigi menawarkan total solusi IT untuk klien mereka yang bergerak di bisnis retail dan F&B , meliputi analisis proses bisnis,pengembangan sistem dan software, pelatihan pengguna , pendampingan pada saat implementasi dan kontrak pemeliharaan sistem secara berkala.

Alamat : Rukan Plaza Pasifik Blok B4 No.81, Jl. Boulevard Barat, Kelapa Gading, RT.18/RW.8, Klp. Gading Bar., Jakarta Utara, Kota Jkt Utara, Daerah Khusus Ibukota Jakarta 14260

Telepon : (021) 45875368

Email : [info@datadigi.com](mailto:info@datadigi.com)



**Gambar 2.1 Logo Perusahaan Datadigi**

#### **2.2 Landasan Teori**

Berikut beberapa landasan teori terkait aplikasi dan dokumen yang akan dibuat dalam penelitian skripsi .

### 2.2.1 Point Of Sale

Pengertian Point Of Sale ( POS ) yaitu kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi. Setiap POS terdiri dari hardware berupa Terminal , PC , Receipt Printer , Cash Drawer , Barcode Scanner dan software dengan beberapa fitur yaitu inventory management , pelaporan , purchasing , customer management , standar keamanan transaksi , return processing . Kedua komponen tersebut di gunakan untuk setiap transaksi . Point Of Sale dapat berupa sebuah *checkout counters* dalam sebuah toko [10] . POS akan menjadi sangat penting dalam dunia bisnis Retail dan F&B , karena POS adalah jembatan keluar masuk nya uang antara pembeli dan penjual . Pembayaran tersebut merupakan indikator bagi pebisnis untuk mengukur pendapatan mereka .

### 2.2.2 Re-Engineering

Reengineering atau rekayasa ulang konsep yang pertama kali di kemukakan oleh Michael Hammer dan James Champy di awal 90an , bernama Business Process Reengineering (BPR) [11] , yang dalam bahasa indonesia di sebut ‘rekayasa ulang proses bisnis’ .

Hammer dan Champy mendefinisikan rekayasa ulang sebagai pemikiran ulang secara mendasar dan perancangan ulang secara radikal atas proses bisnis untuk mendapatkan perbaikan beberapa hal , seperti biaya (*cost*) , kualitas (*quality*) , pelayanan (*service*) , dan kecepatan (*speed*) .

Rekayasa ulang perangkat lunak terbagi menjadi 3 tahapan , antara lain :

1. Reverse Engineering

Yaitu proses menganalisis sistem perangkat lunak yang di teliti , secara singkat reverse engineering adalah sebuah proses pengekstrakan informasi yang ada pada sebuah system untuk memahami alur sistem .

2. Restructuring

Setelah reverse engineering dilakukan dan spesifikasi serta alur sistem sudah diidentifikasi , maka proses selanjutnya ada Restructuring. Pada proses ini untuk mengatur ulang dan merekonstruksi *source code* dan memutuskan apakah akan mempertahankan atau merubah bahasa

pemrograman . Namun , ini tidak mempengaruhi fungsionalitas perangkat lunak yang ada , tetapi untuk meningkatkan performa dan pemeliharaan . Dan juga , pada proses rekonstruksi ini dapat dilakukan penghapusan source code yang sering menyebabkan kesalahan pada perangkat lunak .

### 3. Forward Engineering

Merupakan implementasi dari Reverse Engineering untuk menghasilkan sistem yang baru . Tahapan ini meningkatkan kualitas sistem secara keseluruhan . Rangkaian proses forward engineering adalah :

- a. Penentuan spesifikasi dari sistem yang baru
- b. Merancang sistem yang baru
- c. Pembuatan sistem yang baru
- d. Pengujian

### **2.2.3 Maintainability**

Maintainability dapat di artikan sebagai pemeliharaan suatu sistem yang memungkinkan sistem tersebut rusak, dengan menerapkan konsep Maintainability tentunya kita akan membuat sistem yang akan mudah di pelihara.

### **2.2.4 Android**

Menurut Nasruddin Safaat H , Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux[12]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak.

Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007.

### 2.2.5 Android Studio

Android studio merupakan sebuah Integrated Development Environment (IDE) untuk platform android. Android Studio ini diumumkan pada tanggal Mei 2013 pada Google I/O oleh produk Manager Google , Ellie Powers .

Versi pertama android adalah 0.1 yang dirilis pada Mei 2013 , dan memasuki tahap beta sejak versi 0.8 dirilis pada juni 2014 . Versi stabil yang pertama kali dirilis adalah versi 1.0 sejak Desember 2014 . Sedangkan versi stabil saat ini adalah versi 3.6.1 per-Maret 2020.

### 2.2.6 Design Pattern

Design Pattern adalah metode untuk membantu permasalahan yang berulang dalam pengembangan software, didalam buku “*Design Patterns: Elements of Reusable Object-Oriented Software*” tertulis bahwa “*Design patterns help you choose design alternatives that make a system reusable and avoid alternatives that compromise reusability*”.

Design Pattern bisa juga di sebut sebagai acuan pola dalam pengembangan software, karena kita sebagai software engineering bekerja secara team , agar satu dengan yang lainnya mengerti satu sama lain terkait pola serta arsitektur dalam pengembangan software .

### 2.2.7 Java

Menurut Shalahudin dan Rosa Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan[14]. Java 2 adalah generasi kedua dari Java Platform, sedangkan generasi awalnya adalah *Java Development Kit*.

Menurut Sun Microsystem dalam Sutopo dan Masya (2005) bahasa pemrograman java mempunyai karakteristik sederhana, berorientasi objek, terdistribusi , interpreter, performance tinggi, multithreaded, dan dinamis.

### **2.2.8 Kotlin**

Kotlin adalah bahasa pemrograman berbasis Java Virtual Machine (JVM) yang dikembangkan oleh JetBrains [15]. Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan object oriented (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk backend. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi diatas platform JVM adalah sebagai berikut:

1. Dapat mengatasi NullPointerException yang umumnya terdapat pada Java.
2. Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java [16].
3. Mudah dipelajari
4. Dukungan IDE untuk mempermudah dalam pemrograman

### **2.2.9 Web Service**

Web service adalah untuk sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

Web service bertujuan untuk meningkatkan kolaborasi antar pemogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemograman yang terdapat

di dalamnya. Beberapa alasan mengapa digunakannya Web Service adalah sebagai berikut:

1. Web Service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web Service memiliki kemudahan dalam proses deploymentnya karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web Service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan orientasi.
3. Web service berjalan di port 80 yang merupakan protocol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

#### **2.2.10 API**

API (Application Programming Interface) adalah Sebuah bahasa dan format pesan yang digunakan oleh sebuah aplikasi untuk berkomunikasi dengan: sistem operasi, DBMS (Database Management System), protokol komunikasi, Hardware atau program lain. API digunakan melalui pemanggilan fungsi API pada program. API menyediakan fungsi untuk menghubungkan dengan subroutine yang diperlukan pada eksekusi program. [13]

#### **2.2.11 Bluetooth**

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2.4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping transceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan terbatas. Bluetooth 4.0 adalah teknologi Bluetooth versi terbaru dengan keunggulan utama hemat energy yang resmi diadopsi oleh Bluetooth Special Interest Group (SIG) pada tahun 2010 [17].

Bluetooth 4.0 memiliki beberapa keunggulan yaitu:

1. Konsumsi daya yang lebih kecil sehingga hemat baterai atau listrik
2. Waktu pemakaian yang lebih lama sampai tahunan dengan hanya menggunakan baterai kapasitas kecil (coin cell batteries).
3. Biaya produksi yang rendah.
4. Jarak yang lebih panjang. Mampu menjangkau lebih dari 100 meter.
5. Kecepatan transfer yang lebih baik dengan kemampuan sekitar 1 Mbps.
6. Sistem keamanan yang lebih terkontrol dibandingkan dengan sistem keamanan pada Bluetooth generasi-generasi sebelumnya.

### **2.2.12 UML ( Unified Modelling Language )**

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak[18]. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

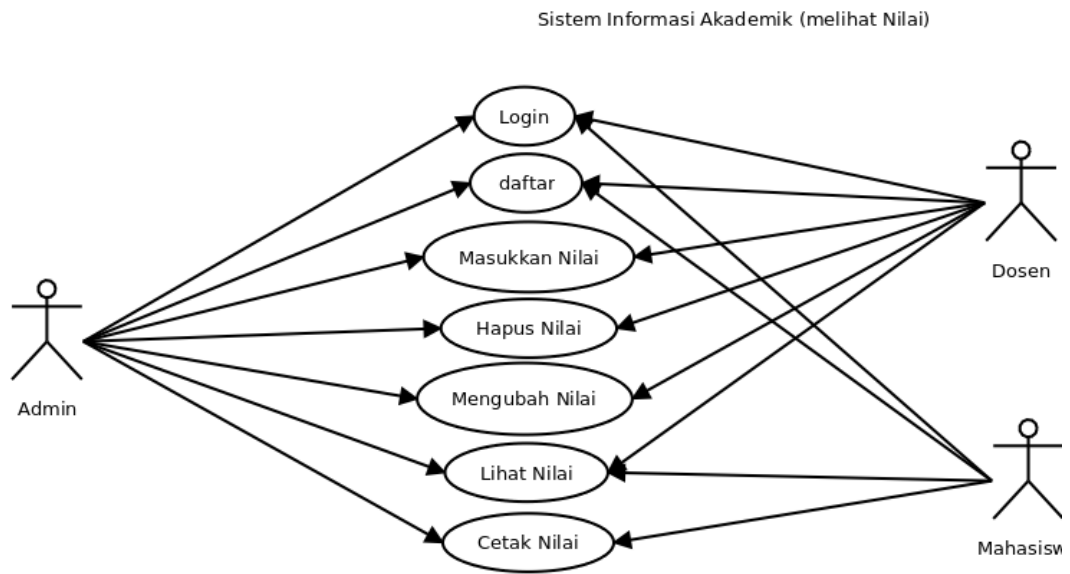
Secara filosofi UML di ilhami oleh konsep yang sudah ada yaitu pemodelan Object Oriented karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik.

### **2.2.13 Usecase Diagram**

Prabowo Pudjo Widodo, menyatakan bahwa use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas[19].

Menurut Pooley dalam Muslihudin [20] mengatakan bahwa model use case dapat dijabarkan dalam diagram use case, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. Komponen pembentuk diagram use case adalah :

- a. Aktor (actor), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. Use case, aktivitas atau sarana yang disiapkan oleh bisnis atau sistem.
- c. Hubungan (link), aktor mana saja yang terlibat dalam use case ini.

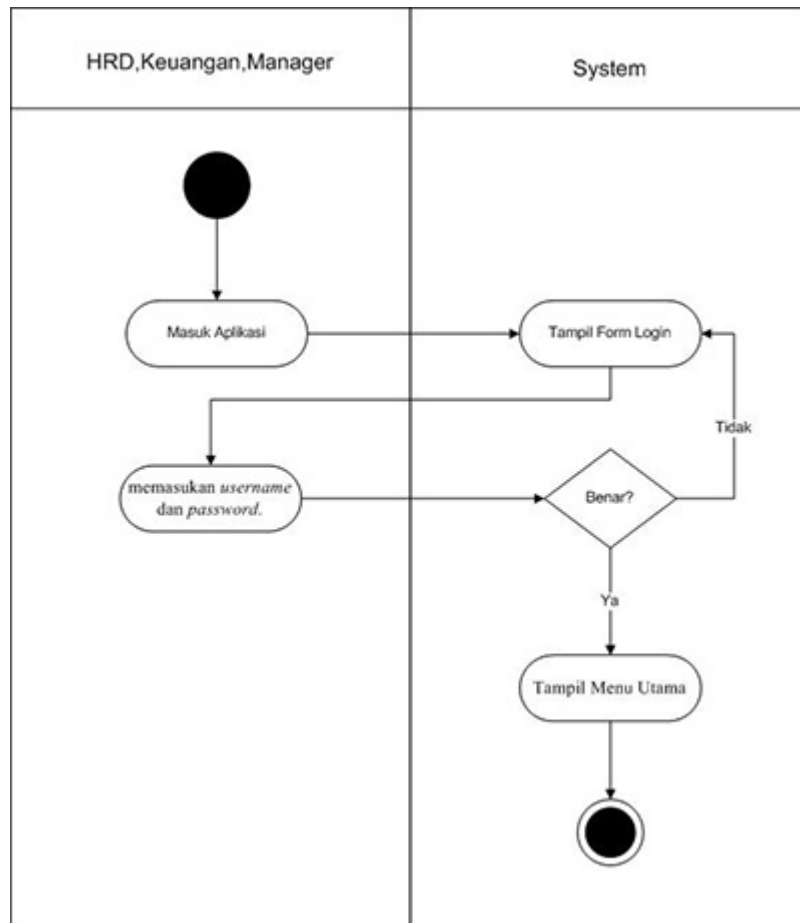


**Gambar 2.2 Contoh Usecase Diagram**

#### 2.2.14 Activity Diagram

Muslihudin mengemukakan bahwa, Diagram aktivitas bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem[20]. Diagram ini terutama penting dalam suatu sistem serta pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antarobjek.



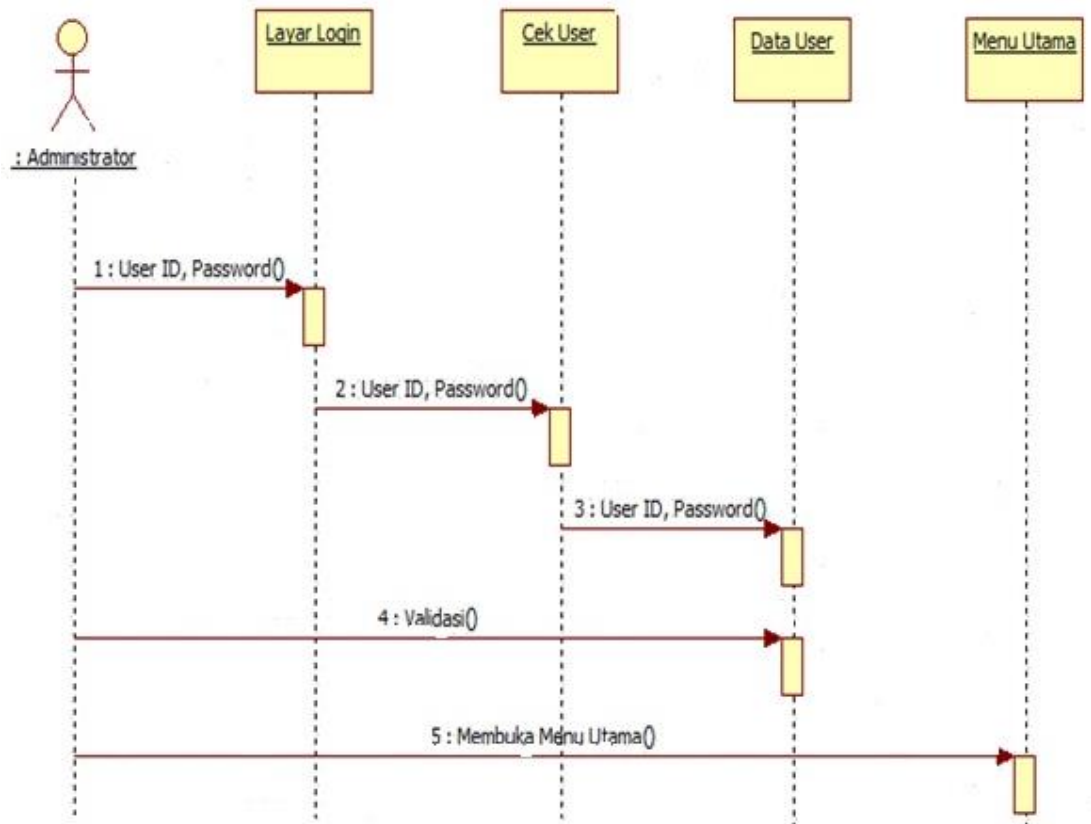


**Gambar 2.3 Contoh Activity Diagram**

### 2.2.15 Sequence Diagram

Muslihudin menyatakan bahwa diagram interaksi dan *sequence* (urutan) bersifat dinamis. Dinamis urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu[20].

*Sequence Diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, kronologi urutan perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.



Gambar 2.4 Contoh Sequence Diagram

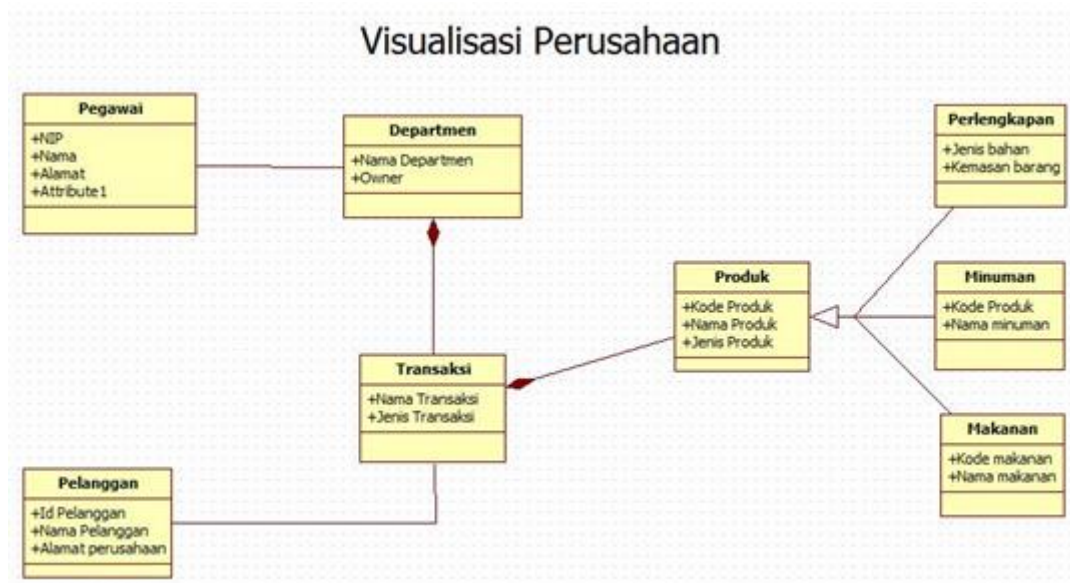
### 2.2.16 Class Diagram

Munawar menyatakan bahwa, Class diagram merupakan himpunan dari objek-objek yang sejenis[21]. 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.

Menurut Whitten, class diagram menggambarkan struktur objek yang terdapat pada sebuah sistem. Diagram ini menunjukkan objek – objek yang terdapat pada suatu sistem dan relasi antar objek – objek tersebut[22]. Class memiliki tiga area pokok yaitu :

- a. Nama, kelas harus mempunyai sebuah nama.
- b. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.

- c. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.



Gambar 2.5 Contoh Class Diagram

### 2.2.17 Maintainability Index

Maintainability Index merupakan sebuah metrik yang mengukur perangkat lunak untuk mengembangkan perangkat lunak tersebut. Maintainability Index (MI) memudahkan pengembangan jika kode sumbernya hendak dirubah. Pada umumnya, indeks pada MI dihitung terdiri dari Lines of Code, Cyclomatic Complexity dan Halstead volume. Namun, pada perangkat lunak android, ada beberapa spesial fitur spesial yang tidak terdapat pada OO[23]. Oleh karena itu, indeks matriks yang ditentukan untuk perhitungan pada MI untuk Android berbeda. Berikut ini adalah metriks tersebut:

METRIKS	DEFINISI
CL_WMC	JUMLAH WEIGHTED METHOD TIAP KELAS
IN_BASES	JUMLAH KELAS YANG INHERIT SECARA LANGSUNG
CU_CDUSED	JUMLAH CLASS YANG DIGUNAKAN SECARA LANGSUNG OLEH KELAS INI
CL_CMOF	TOTAL JUMLAH BARIS DALAM KODE
CL_STAT	JUMLAH STATEMENT YANG DAPAT DI EKSEKUSI
CL_DATA	JUMLAH TOTAL <u>ATRIBUTES YANG TERDAPAT</u> TIAP KELAS
CL_FUN	JUMLAH TOTAL <u>FUNCTION YANG TERDAPAT</u> TIAP KELAS
CL_FUN_PUB	JUMLAH PUBLIC FUNCTION YANG ADA DITIAP KELAS
CU_CDUSERS	JUMLAH KELAS YANG MENGGUNAKAN KELAS INI SECARA LANGSUNG
IN_NOC	JUMLAH CHILDREN DARI KELAS INI
CL_DATA_PUB	JUMLAH TOTAL ATRIBUTES PUBLIC YANG TERDAPAT TIAP KELAS
NOP	JUMLAH REQUESTED PERMISSION (<-USED PERMISSION->)

**Gambar 2.6 Metriks MI**

Setiap kelas dalam sebuah fitur, akan dicari metriks tersebut. Dan bila setiap kelas sudah mendapatkan metriks, seluruh unsur metriks masing-masing akan dirata-ratakan, kemudian hasilnya akan dimasukan ke dalam rumus di bawah ini:

$$MI = NOP + IN\_NOC + IN\_BASES + 2*CL\_FUN + CL\_CMOF + CL\_DATA + CL\_STAT + CL\_FUN\_PUB + CU\_CDUSERS + CL\_DATA\_PUB + 2*CU\_CDUSD + 2*CL\_WMC$$

Nilai Maintainability Index	Klasifikasi
$MI > 85$	<i>Highly Maintainable</i>
$MI 65 \leq 85$	<i>Moderately Maintainable</i>
$MI < 65$	<i>Difficult to Maintain</i>