



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian *Customer Relationship Management* (CRM)

Menurut Kundre, et.al, (2013:9-10),”*Customer Relationship Management* (CRM) merupakan jenis manajemen secara khusus membahas teori mengenai penanganan hubungan antara perusahaan dengan pelanggannya, yang mana tujuannya untuk meningkatkan hubungan dengan tiap pelanggan demi mencapai pertumbuhan yang sehat”.

Menurut Handojo, et al (2010:1-2), Adapun beberapa tahapan yang digunakan *Customer Relationship Management* (CRM) yaitu,

1. Mendapatkan pelanggan baru (*acquire*). Pelanggan baru didapatkan dengan memberikan kemudahan pengaksesan informasi, inovasi baru, dan pelayanan yang menarik.
2. Meningkatkan hubungan dengan pelanggan yang telah ada (*enhance*). Perusahaan berusaha menjalin hubungan dengan pelanggan melalui pemberian pelayanan yang baik terhadap pelanggannya (*customer service*). Penerapan *cross selling* atau *up selling* pada tahap kedua dapat meningkatkan pendapatan perusahaan dan mengurangi biaya untuk memperoleh pelanggan (*reduce cost*).
3. Mempertahankan pelanggan (*retain*). Tahap ini merupakan usaha mendapatkan loyalitas pelanggan dengan mendengarkan pelanggan dan berusaha memenuhi keinginan pelanggan.

Dari kedua pendapat tersebut dapat disimpulkan bahwa *Customer Relationship Management* (CRM) merupakan suatu penerapan dalam jenis manajemen yang membahas tentang penanganan hubungan antara perusahaan dengan pelanggannya, bertujuan agar dapat meningkatkan hubungan yang baik pada pelanggan dan meningkatkan nilai perusahaan di mata para pelanggannya. Dimana tahapan-tahapan yang digunakan *Customer Relationship Management* (CRM) terdiri dari mendapatkan pelanggan baru (*acquire*), meningkatkan



hubungan dengan pelanggan yang telah ada (*enhance*), dan mempertahankan pelanggan (*retain*).

Menurut Dyantina, et al (2012:518-519), dari sudut pandang arsitektur, kerangka CRM dapat diklasifikasikan ke dalam tiga komponen utama, antara lain:

1. Operasional CRM

Operasional CRM dikenal sebagai *front office* perusahaan. Komponen CRM ini berperan dalam interaksi dengan pelanggan. Operasional CRM mencakup proses otomatisasi yang terintegrasi dari keseluruhan proses bisnis, seperti otomatisasi pemasaran, dan pelayanan. Salah satu penerapan CRM yang termasuk dalam kategori operasional CRM adalah dalam bentuk aplikasi web. Melalui web, suatu perusahaan dapat memberikan pelayanan kepada pelanggan.

2. Analitikal CRM

Analitikal CRM dikenal sebagai *back office* perusahaan. Komponen CRM ini berperan dalam memahami kebutuhan pelanggan. Analitikal CRM berperan dalam melaksanakan analisis pelanggan dan pasar, seperti analisis trend pasar dan analisis kebutuhan dan perilaku pelanggan. Data yang digunakan pada CRM analitik adalah data yang berasal dari CRM operasional.

3. Kolaborasi CRM

Komponen kolaborasi CRM meliputi e-mail, *personalized publishing*, *ecomunities*, dan sejenisnya yang dirancang untuk interaksi antara pelanggan dengan perusahaan. Tujuan utamanya adalah memberikan nilai tambah dan memperluas loyalitas pelanggan ke pelanggan lain yang masih belum berada di level kesetiaan pelanggan. *Collaborative CRM* juga mencakup pemahaman atau kesadaran bahwa pelanggan yang setia dapat menjadi magnet bagi pelanggan lain.



2.1.2 Metodologi Agile

Menurut Muslim, et al (2014:30), “Metode *agile* adalah metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun”.

Menurut Suryantara (2017:22), “Metodologi *agile* merupakan metodologi yang cepat”. Beberapa metodologi agile yang berkembang saat ini antara lain:

- 1) *Extreme Programming* (XP)
- 2) *Scrum Methodology*
- 3) *Crystal Family*
- 4) *Dynamic Systems Development Method* (DSDM)
- 5) *Adaptive Software Development* (ASD)
- 6) *Feature Driven Development* (FDD)

Dari kedua pendapat tersebut dapat disimpulkan bahwa metodologi *agile* merupakan metodologi pengembangan perangkat lunak didasarkan pada prinsip-prinsip yang sama. Metodologi *agile* ini dapat melakukan suatu pengembangan sistem secara jangka pendek dengan memerlukan adaptasi yang cepat dari pengembang terhadap perubahan dalam bentuk apapun. Sehingga metodologi *agile* ini telah berkembang menjadi beberapa metodologi seperti *Extreme Programming* (XP), *Scrum Methodology*, *Crystal Family*, *Dynamic Systems Development Method* (DSDM), *Adaptive Software Development* (ASD), dan *Feature Driven Development* (FDD).

2.1.2.1 Metode *Extreme Programming*

Menurut Prabowo (2013:2), “*Extreme programming* (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan – perubahan *requirement* yang sangat cepat”.



Menurut Pratama (2017:93), “*Extreme Programming (XP)* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih *adaptif* dan *fleksibel*”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa metode *extreme programming (XP)* merupakan metode pengembangan perangkat lunak yang ringan dan termasuk salah satu metode *agile*. *Extreme Programming (XP)* lebih cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium. Sehingga metode *extreme programming (XP)* lebih mengedepankan proses pengembangan yang lebih *responsive* terhadap kebutuhan.

2.1.3 UML (*Unified Modelling Language*)

Menurut Rosa dan Shalahuddin (2013:133), “*UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Menurut Pratama (2013:48), “*Unified Modelling Language (UML)* adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek”.



Gambar 2.1 Logo UML

Sumber : Pratama (2013)

Dari kedua pendapat tersebut dapat disimpulkan bahwa *unified modelling language (UML)* merupakan alat atau pemodelan untuk mendukung pengembangan sistem yang berorientasi objek (OOP), berfungsi untuk untuk



pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

2.1.3.1 Pengertian *Use Case Diagram*




Menurut Pratama (2013:48), “*Use case diagram* menggambarkan aliran kegiatan dan proses bisnis yang dilakukan oleh pengguna (aktor)”.

Menurut Muslihudin, et al (2016:64), “Diagram *use case* bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (Suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

Dari kedua pendapat tersebut dapat disimpulkan bahwa diagram *use case* merupakan suatu diagram yang bersifat statis, dimana menggambarkan aliran kegiatan dan proses bisnis yang dilakukan oleh pengguna (*actor*) yang memperlihatkan himpunan *use case* dan aktor-aktor (Suatu jenis khusus dari kelas).

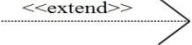
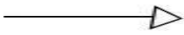
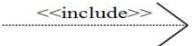
Menurut Rosa dan Shalahuddin (2013 : 162), Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1 Simbol *Use Case Diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|--------------------|---|
| 1 |  | <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. |
| 2 |  | <i>Actor</i> | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. |
| 3 |  | <i>Association</i> | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |



Tabel 2.1 Simbol *Use Case* Diagram

| | | | |
|---|---|-----------------------|--|
| 4 |  | <i>Extend</i> | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. |
| 5 |  | <i>Generalization</i> | Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lain. |
| 6 |  | <i>Include</i> | 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 sebagai syarat dijalankan <i>use case</i> . |

Sumber: Rosa dan Shalahuddin (2013 : 162)

2.1.3.2 Pengertian *Activity Diagram*

Menurut Triadini dan Suardika (2012:37), “*Activity Diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut”.






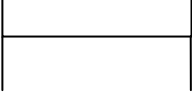
Menurut Rosa dan Shalahuddin (2013:161), “*Diagram aktivitas* atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa diagram *activity* merupakan diagram yang bersifat dinamis ,dimana menggambarkan struktur kelas yang terdapat pada perangkat lunak yang akan dibangun. Sehingga diagram ini sangat penting dalam suatu sistem serta pemodelan fungsi–fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.



Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.2 Simbol *Activity Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|--------------|--|
| 1 |  | Status Awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2 |  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 |  | Decision | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | Join | Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu. |
| 5 |  | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6 |  | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber: Rosa dan Shalahuddin (2013 : 162)

2.1.3.3 Pengertian *Class Diagram*

Pratama (2013:48), “*Class diagram* menggambarkan struktur kelas yang terdapat pada perangkat lunak yang akan dibangun”.

Menurut Muslihudin, et al (2016:62),” Diagram kelas bersifat statis . Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka–antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

Dari kedua pendapat tersebut dapat disimpulkan bahwa diagram *class* merupakan suatu diagram yang bersifat statis. Dimana pada umumnya dijumpai pada pemodelan sistem berorientasi objek, untuk menggambarkan struktur kelas yang terdapat pada perangkat lunak yang akan dibangun.



Menurut Rosa dan Shalahuddin (2013 : 162), adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut;

Tabel 2.3 Simbol *Class* Diagram

| No. | Gambar | Nama | Keterangan |
|-----|--------|-----------------------------|--|
| 1 | | Kelas | Kelas pada struktur sistem. |
| 2 | | <i>Interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| 3 | | <i>Association</i> | Relasi antarkelas dengan makna umum. |
| 4 | | <i>Directed Association</i> | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
| 5 | | Generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus). |
| 6 | | <i>Dependency</i> | Relasi antarkelas dengan makna kebergantungan antarkelas. |
| 7 | | <i>Aggregation</i> | Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>). |

Sumber: Rosa dan Shalahuddin (2013 : 162)




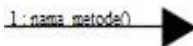
2.1.3.4 Pengertian *Sequence Diagram*

Menurut Muslihudin, et al (2016:62), “Diagram interaksi dan *sequence* (urutan) bersifat dinamis. Dinamis urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

Menurut Rosa dan Shalahuddin (2013:161), “Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Adapun simbol-simbol yang digunakan dalam *sequence* diagram adalah sebagai berikut:

Tabel 2.4 Simbol *Sequence Diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|-----------------|--|
| 1 |  | Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. |
| 2 |  | Lifeline | Menyatakan kehidupan suatu objek. |
| 3 |  | Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
| 4 |  | Pesan tipe call | Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. |

Sumber: Rosa dan Shalahuddin (2013 : 162)

2.1.4 Pengertian *Apache Cordova*

Menurut Sopingi (2016),” *Apache cordova* adalah *framework open source* untuk pengembangan aplikasi mobile tanpa menggunakan bahasa pemrograman native pada platform mobile. Aplikasi dieksekusi menggunakan standart API yang tersedia untuk mengakses data, device sensor atau status jaringan pada device. *Apache cordova* dapat digunakan untuk mengembangkan aplikasi pada beberapa



platform, pengembangan web yang didistribusikan kedalam *mobile device* dan menggabungkan komponen aplikasi bahasa pemrograman *native* dengan WebView yang dapat mengakses API pada perangkat *mobile*. (Sopingi, 2016:4) diacu dalam Jurnal Duta, Vol.10, No.1).

Menurut Tompoh , et al (2016), “*Apache cordova* atau disebut saja *cordova* adalah *open-source framework* untuk pengembangan aplikasi *mobile*, yang memungkinkan pengembang menggunakan teknologi web standar seperti HTML5, CSS3, dan JavaScript untuk pengembangan *crossplatform*. Aplikasi *cordova* tetap dikemas sebagai paket aplikasi menggunakan *Software Development Kit* (SDK) dari platform masing-masing. Maksudnya jika paket dijalankan pada android maka pengemasannya harus menggunakan SDK android, demikian pula dengan platform yang lainnya. *Cordova* merupakan satu set *Application Programming Interface* (API) perangkat yang memungkinkan pengembang aplikasi *mobile* untuk mengakses fungsi perangkat asli (*native*) seperti kamera atau *accelerometer* dengan menggunakan bahasa *javascript*. (Tompoh , et al, 2016:2) diacu dalam E-journal Teknik Informatika, Vol 9, No 1).

Dari pendapat tersebut dapat disimpulkan bahwa *apache cordova* merupakan *framework open source* untuk mengembangkan aplikasi *mobile* tanpa menggunakan bahasa pemrograman *native* pada platform *mobile*. Sehingga dapat memungkinkan pengembangan ini menggunakan teknologi web standar seperti HTML5, CSS3, dan *JavaScript* untuk pengembangan *crossplatform*.

2.1.5 Pengertian *Ionic Framework*

Menurut Sopingi (2016), “*Ionic* adalah kerangka ponsel HTML5 dengan fokus pada kinerja yang memanfaatkan akselerasi hardware dan tidak memerlukan pihak ketiga seperti *JavaScript library*”. (Sopingi, 2016:4) diacu dalam Jurnal Duta, Vol.10, No.1).

Menurut Tompoh , et al (2016), “*Ionic Framework* adalah kerangka kerja yang dibangun untuk mengembangkan aplikasi *hybridmobile* dengan menggunakan teknologi web. *Ionic* menyediakan komponen-komponen *mobile* seperti *button*,



header-footer, cards, dan lain sebagainya kemudian menyediakan *css* standar dan menggunakan *javascript* untuk menjalankan fungsi-fungsi aplikasi, untuk *javascript* sendiri *ionic* memakai teknologi *angular.js*". (Tompoh , et al, 2016:2) diacu dalam E-journal Teknik Informatika, Vol 9, No 1).

Dari pendapat tersebut dapat disimpulkan bahwa *ionic Framework* merupakan kerangka ponsel HTML5 dengan fokus pada kinerja yang memanfaatkan akselerasi hardware dan tidak memerlukan pihak ketiga seperti *JavaScript library*. Dimana *Ionic* menyediakan komponen-komponen mobile seperti *button, header-footer, cards*, dan lain sebagainya kemudian menyediakan *css* standar dan menggunakan *javascript* untuk menjalankan fungsi-fungsi aplikasi, untuk *javascript* sendiri *ionic* memakai teknologi *angular.js*.

2.1.6 Pengertian Java

Menurut Wardhani (2013) java adalah sebuah bahasa pemrograman yang populer dikalangan para akademisi dan praktisi komputer. Java pertama kali dikembangkan untuk memenuhi kebutuhan akan sebuah bahasa komputer yang ditulis satu kali dan dapat dijalankan dibanyak *system* komputer berbeda tanpa perubahan kode berarti. Pada umumnya, para pakar pemrograman berpendapat bahwa bahasa Java memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan. (Wardhani , 2013:474) diacu dalam Jurnal Teknika, Vol. 5 No.2).

Menurut Barri (2015) java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. (Barri ,2015:2) diacu dalam E-journal Teknik Elektro dan Komputer).

Dari pendapat tersebut dapat disimpulkan bahwa *java* merupakan suatu bahasa pemrograman yang dikembangkan untuk memenuhi kebutuhan sebuah bahasa komputer yang dapat dijalankan di di berbagai komputer termasuk telepon genggam.



2.1.7 Pengertian Xampp

Komputer Wahana (2014:72), “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP Server, *PhpMyAdmin* dan berbagai pustaka bantu lainnya”.

Abdulloh (2016:7), “XAMPP adalah salah satu paket *installer* yang berisi Apache yang merupakan *web server* tempat menyimpan file-file yang dibutuhkan *website*, dan *PhpMyAdmin* sebagai aplikasi yang digunakan untuk perancangan *database MySQL*”.

Dari pendapat tersebut dapat disimpulkan bahwa XAMPP adalah salah satu paket *installer* yang berisi Apache yang merupakan *web server* tempat menyimpan file-file yang dibutuhkan *website*, dan *PhpMyAdmin* sebagai aplikasi yang digunakan untuk perancangan *database MySQL*.

2.2 Teori Khusus

2.2.1 Pengertian Aplikasi

Asropudin (2013:7), “Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel*”.

Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Ms-word*, *Ms-excel*”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa aplikasi merupakan *software* yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu.



2.2.2 Pengertian Reservasi

Menurut Aprilia, et al (2017), “Reservasi sendiri merupakan suatu permintaan seorang tamu agar memperoleh sejumlah kamar yang diinginkan, yang dilakukan beberapa waktu sebelumnya lewat beberapa sumber dan berbagai cara pemesanan dengan tujuan memastikan bahwa tamu tersebut akan mendapatkan kamar pada saat mereka datang atau check-in”. (<https://media.neliti.com/media/publications/87777-ID-analisis-sistem-informasi-reservasi-hote.pdf>, diakses 5 april 2017).

Menurut Wijaya, et al (2013), “*Reservation* yaitu pemesanan fasilitas yang diantaranya akomodasi, *meal*, *seat* pertunjukan pada, pesawat terbang, kereta api, bus, hiburan, *night club*, *discotheque* dan sebagainya”. (<https://media.neliti.com/media/publications/74758-ID-none.pdf>, diakses 5 april 2018).

Dari kedua pendapat tersebut dapat disimpulkan bahwa reservasi merupakan permintaan seorang tamu dalam melakukan pemesanan sejumlah kamar yang diinginkan ataupun fasilitas.

2.2.3 Pengertian Hotel

Menurut Chair (2017:2), “Hotel merupakan suatu jenis akomodasi yang menggunakan sebagian atau seluruh bangunan untuk menyediakan jasa pelayanan penginapan, makan dan minum serta jasa lainnya bagi umum, yang dikelola secara komersial serta memenuhi ketentuan persyaratan yang ditetapkan didalam keputusan pemerintah”.

Menurut Utama (2016:42), “Hotel merupakan sejenis akomodasi yang menyediakan fasilitas dan pelayanan penginapan, makan, dan minum serta jasa-jasa lainnya untuk umum yang tinggi untuk sementara waktu dan di kelola secara professional”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa hotel merupakan suatu jenis akomodasi menggunakan sebagian atau seluruh bangunan untuk menyediakan jasa pelayanan dan fasilitas bagi masyarakat umum, dimana



masyarakat umum yang memiliki kebutuhan dalam travelling ataupun urusan bisnis.

2.2.4 Pengertian Android

Menurut Suprayadi (2017:2), “Android merupakan sebuah sistem operasi perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Beberapa pengertian lain dari android, yaitu;

1. Merupakan platform terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli oleh Google inc dari Android Inc.

Menurut Supardi (2015:1), “Android merupakan sistem operasi berbasis Linux yang digunakan untuk telepon seluler (*mobile*), seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).

Dari pendapat tersebut dapat disimpulkan bahwa android merupakan sistem operasi berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi, biasanya digunakan untuk telepon seluler (*mobile*), seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).