# **BAB II KAJIAN PUSTAKA**

## Fiber Optik

Serat optik merupakan kabel atau bisa yang disebut sebagai media transmisi yang terbuat dari kaca atau plastic yang sangat halus dan lebih kecil dari sehelai rambut, dan dapat digunakan untuk mentransmisikan sinyal cahaya dari suatu tempat ke tempat lain. Sumber cahaya yang digunakan biasanya adalah laser atau LED. Kabel ini berdiameter lebih kurang 120 mikrometer. Cahaya yang ada di dalam serat optik tidak keluar karena indeks bias dari kaca lebih besar daripada indeks bias dari udara, karena laser mempunyai spektrum yang sangat sempit. Alasan kecepatan transmisi pada serat optik sangat tinggi sehingga sangat bagus digunakan sebagai saluran komunikasi. Saat ini teknologi serat optik memiliki attenuation kurang dari 20 decibels (dB)/km. Hal ini dipengaruhi oleh proses penyerapan (Absorption), yakni kehilangan cahaya yang disebabkan adanya kotoran di dalam serat optik, lalu pada saat proses penyebaran (Scattering) dan pada saat kehilangan radiasi (radiative losses). Kabel serat optik terdiri dari 2 bagian utama, yaitu cladding dan core. Yang di mana cladding merupakan selubung dari inti (core). Dan core merupakan inti dari lapisan fiber optik.

Berikut merupakan 3 jenis serat optik yang sering digunakan, yaitu (Senjaya, Alvino, 2014) :

1. Single mode

Merupakan serat optik dengan core yang sangat kecil, pada mode ini cahaya yang masuk ke dalam serat optik tidak terpantul –pantul ke dinding ( cladding) yang disebabkan oleh diameter inti yang sangat sempit mendekati panjang gelombang. Pelemahan yang dihasilkan pada mode ini yaitu kurang dari 0.35 dB per kilometer yang di mana untuk hasil attenuation ini dapat memungkinkan kecepatan yang sangat tinggi dari jarak yang sangat jauh. ITU-T G.652dan G.657 merupakan standar terbaru untuk kabel ini.

1. Multi mode

Merupakan serat optik dengan diameter core yang agak besar yang membuat laser di dalamnya akan terpantul-pantul di dinding (cladding) yang dapat menyebabkan berkurangnya bandwidth dari serat optik jenis ini.

1. Grade Indeks

Merupakan serat optik yang memiliki diameter core yang besar dan mempunyai cladding yang bertingkat indeks biasnya sehingga dapat menambah bandwidth. Diameter core berukuran 30-60mm, cladding 100-150mm. Merupakan penggabungan kabel fiber single mode dan multimode. Transmisi jarak 10-20 km.

## Arsitektur Jaringan Fiber Optik Secara Umum

Fiber To The X (FTTX) merupakan istilah umum yang sering digunakan untuk beberapa arsitektur jaringan fiber optik dalam dunia telekomunikasi. Berdasarkan modus aplikasinya, beberapa arsitektur jaringan lokal akses fiber tersebut dibagi menjadi Fiber To The Building (FTTB), Fiber To The Zone (FTTZ), Fiber To The Curb (FTTC), Fiber To The Home (FTTH).

1. Fiber To The Building (FTTB)

Titik Konversi Optik (TKO) terletak di dalam gedung dan biasanya terletak pada ruang telekomunikasi di basement namun juga dimungkinkan diletakkan pada beberapa lantai di gedung tersebut. Terminal pelanggan dihubungkan dengan TKO melalui kabel tembaga indoor. FTTB dalam diterapkan bagi pelanggan bisnis di gedung-gedung bertingkat atau bagi pelanggan perumahan di apartemen.

1. Fiber To The Zone (FTTZ)

TKO terletak di suatu tempat di luar bangunan, baik di dalam kabinet dengan kapasitas besar. Terminal pelanggan dihubungkan dengan Titik Konversi Optik (TKO) melalui kabel tembaga hingga beberapa kilometer. FTTZ umumnya diterapkan pada daerah perumahan yang letaknya jauh dari sentral atau infrastruktur duct pada arah yang bersangkutan, sudah tidak memenuhi lagi untuk ditambahkan dengan kabel tembaga.

1. Fiber To The Curb (FTTC)

Titik Konversi Optik (TKO) terletak di suatu tempat di luar bangunan, di dalam kabinet dan diatas tiang dengan kapasitas lebih kecil. Terminal pelanggan dihubungkan dengan TKO memalui kabel tembaga hingga beberapa ratus meter. FTTC dapat diterapkan bagi pelanggan bisnis yang letaknya berkumpul di suatu area terbatas namun tidak berbentuk gedung-gedung bertingkat atau bagi pelanggan perumahan yang pada waktu dekat akan menjadi pelanggan jasa hiburan.

1. Fiber To The Home(FTTH)

Fiber to the Home (disingkat FTTH) merupakan suatu format pengantaran isyarat optik dari pusat penyedia (provider) ke kawasan pengguna dengan menggunakan serat optic sebagai medium pengantaran. Perkembangan teknologi ini tidak terlepas dari kemajuan perkembangan teknologi serat optik yang dapat menggantikan penggunaan kabel konvensional. Dan juga didorong oleh keinginan untuk mendapatkan layanan yang dikenal dengan istilah Triple Play Services yaitu layanan akan akses internet yang cepat, suara (jaringan telepon, PSTN) dan video (TV Kabel) dalam satu infrastruktur pada unit pelanggan (Ramadhan, dkk., 2012).

## Power Link Budget

Power link budget dihitung sebagai syarat agar link yang digunakan dayanya tidak melebihi batas ambang dari daya yang di butuh kan. Standar yang ditentukan sesuai dengan ITU-T G.984sebesar 28 dB. Pada standar reality perhitungan Power Link Budget di lapangan, perangkat dapat bekerja dengan baik pada nilai 25 dB (Telkom, 2015). Untuk menghitung power link budget dapat dihitung dengan rumus:

Bentuk persamaan untuk perhitungan margin daya adalah :

Keterangan :

Pt = Daya keluaran sumber optik ( dBm)

Pr = Sensitivitas daya maksimum detektor ( dBm)

SM = Safety margin, berkisar 6-8 dB

αtot = Redaman Total sistem (dB)

L = Panjang serat optik ( Km)

αc= Redaman Konektor (dB/buah)

αs = Redaman sambungan ( dB/sambungan)

αserat= Redaman serat optik ( dB/ Km)

Ns= Jumlah sambungan

Nc= Jumlah konektor

Sp= Redaman Splitter(dB)

Margin daya disyaratkan harus memiliki nilai lebih dari 0 (nol), margin daya adalah daya yang masih tersisa dari power transmit setelah dikurangi dari loss selama proses mentransmisikan, pengurangan dengan nilai safety margin dan pengurangan dengan nilai sensitifitas receiver.

## Android

Android adalah sebuah sistem operasi perangkat mobile berbasis Linux yang mencangkup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Kelebihan Android

1. Switching dan multitasking yang lebih baik Android sangat mendukung multitasking aplikasi, kini hal tersebut kembali ditingkatkan. Dalam Honey comb pengguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah icon pada system bar.
2. Kapasitas yang lebih baik untuk beragam widget Kapabilitas terhadap beragam widget dijanjikan bakal makin memanjakan para penggunanya. Contohnya widget untuk email Gmail yang dipamerkan Google, pengguna tidak perlu membuka aplikasi Gmail untuk melihat isi di dalamnya.
3. Peningkatan kemampuan copy-paste Beberapa seri Android terdahulu memang sudah bisa melakukan copy-paste, namun beberapa pengguna masalah pemilihan teks yang agak sulit. Kini hal tersebut coba diselesaikan, selain copy-paste Google juga menambah share it pada teks yang diseleksi.

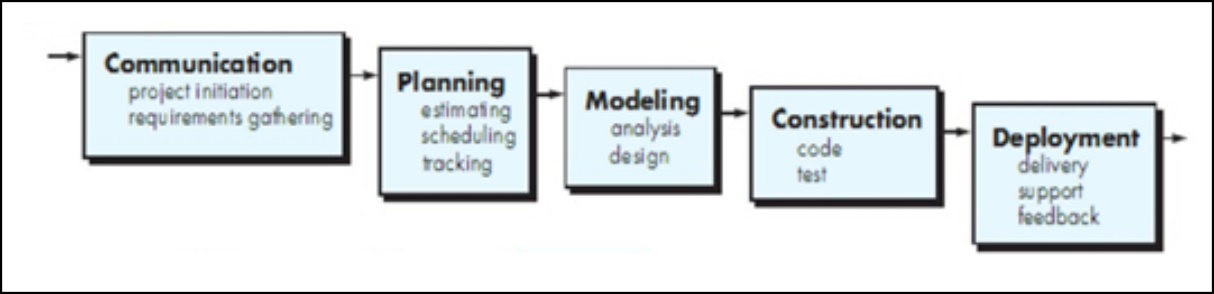
Kekurangan Android

Koneksi Internet yang terus menerus. Kebanyakan ponsel Android memerlukan koneksi internet yang simultan atau terus menerus aktif, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan dan batrai yang boros karena GPRS yang terus menyala.

## Model Waterfall

Menurut Pressman (2015 : 42), metode *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah *“Linear Sequential Model”*. Model ini sering disebut juga dengan *“classic life cycle”* atau metode waterfall. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Fase – fase dalam *Waterfall* *Model* menurut Pressman sebagai berikut :



**Gambar 2.2** Ilustrasi Model *Waterfall (***Sumber** :Pressman, (2015 : 42))

1. ***Communication* (*Project Initiation & Requirements Gathering*)**

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel dan internet.

1. ***Planning* (*Estimating, Scheduling, Tracking*)**

Tahap berikutnya adalah tahapan *Planning /* perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan dan *tracking* proses pengerjaan sistem.

1. ***Modeling* (*Analysis & Design*)**

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface* dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

1. ***Construction* (*Code & Test*)**

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

1. ***Deployment* (*Delivery, Support, Feedback*)**

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software* dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

Metode *waterfall* dianggap pendekatan yang lebih cocok digunakan untuk proyek pembuatan sistem baru dan juga pengembangan *software* dengan tingkat resiko yang kecil serta waktu pengembangan yang cukup lama. Tetapi salah satu kelemahan paling mendasar adalah menyamakan pengembangan *hardware* dan *software* dengan meniadakan perubahan saat pengembangan. Padahal, *error* diketahui saat *software* dijalankan dan perubahan-perubahan akan sering terjadi.

## Unified Modelling Language (UML)

*Unified Modelling Language* (*UML*) merupakan sistem arsitektur yang bekerja dalam *OOAD* (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengonstruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*, dapat berupa model, *deskripsi*, atau *software*) yang terdapat dalam sistem *software.* (Fowler, 2015)

*UML* merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu *Booch, OMT* (*Object Modeling Technique*), dan *OOSE* (*Object-Oriented Software Engineering*). (Munawar. 2015)

Dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini. (Nugroho, 2013)

Tujuan *UML* diantaranya adalah:

1. Memberikan model yang siap pakai, bahasa pemodelan *visual* yang *ekspresif* untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan
4. ***Use Case Diagram***

*Use case diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian *use case diagram*. *Use Case* Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili *user* atau sistem lain yang berinteraksi dengan sistem yang dimodelkan. (Connolly, 2010)

**Tabel 2. 1** Tipe Relasi pada Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | *Deskripsi* dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

*(*Connolly, 2010*)*

1. ***Activity* Diagram**

*Activity* Diagram atau Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Sebuah *activity diagram* memiliki: (Connolly, 2010)

Tabel 2. 2Tipe Relasi pada *Activity Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Penjelasan** |
|  | *Initial State*  Mempresentasikan dimulainya alur kerja suatu sistem dalam *activity diagram*. |
| Action State | *Action State*  Sebuah *state* yang menggambarkan eksekusi dari aksi *atomic*. |
| ► | *Transition Between Activities*  Mengidentifikasikan bahwa suatu objek dari *state* pertama akan menampilkan aksi-aksi tertentu dan memasuki *state* kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya. |
|  | *Decision Point*  Menentukan kapan alur dalam aktivitas menjadi bercabang. |
|  | *Fork*  Adanya percabangan paralel dari aktivitas |
|  | *Final State*  Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam *activity diagram.* |

*(*Connolly, 2010*)*

1. ***Sequence Diagram***

Merupakan diagram yang menunjukkan aliran fungsionalitas dalam *use case*. *Sequence* adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *use case* dan *message* atau pesan-pesannya. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress *vertical.* (Connolly, 2010)

**Tabel 2. 3**Tipe Relasi pada *Sequance Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Elemen** | **Keterangan** |
| A screenshot of a social media post  Description automatically generated | Objek | * Komponen yang menjadikan sebuah objek dalam membuat diagram |
|  | *Stimulus* | * Untuk menandakan hubungan komunikasi antar objek |
|  | *Self Stimulus* | * Fungsi sama dengan *Stimulus,* tetapi pesan yang disampaikan dikirimkan untuk objek itu sendiri |
|  | *Focus Control* | * Sebagai tempat untuk hasil *input* atau *output* dari sebuah proses yang dilakukan oleh objek ataupun aktor yang ada dalam sistem |
| *(*Connolly, 2010*)* |  |  |

***Class* Diagram**

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama.Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling popular di *UML*. (Connolly, 2010)

**Tabel 2. 4**Tipe Relasi pada *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Elemen** | **Keterangan** |
| A screenshot of a cell phone  Description automatically generated | Kelas | * Simbol untuk membangun sebuah pemrograman dengan objek * Terdiri 3 bagian, bagian **atas** adalah nama kelas, bagian **tengah** adalah atribut, dan bagian **bawah** adalah metode dari kelas tersebut |
|  | Garis Asosiasi | * Simbol yang menggambarkan adanya hubungan antara satu kelas dengan kelas lainnya |
|  | Generalisasi | * Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan |

*(*Connolly, 2010*)*

*Unified Modeling Language (UML*) adalah bahasa pemodelan umum yang digunakan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi artifak dari software system. UML bukanlah sebuah standar proses pengembangan dalam metode pengembangan sistem tertentu, namun pada umumnya UML dipakai dalam memodelkan sistem yang dibangun berbasiskan objek.

## MySQL

MySQL diciptakan di Negara Swedia oleh perusahaan MySQL AB adapun masing-masing nama yang berjasa dalam menciptakan MySQL adalah David Axmark, Allan Lasson, dan Michael “*monty*” Widenius. Perangkat lunak ini tersebar luas secara garis besar karena memiliki lisensi *GNU General Public License****.*** Sampai sekarang, tercatat ada beberapa bahasa pemrograman yang popular yang bisa bersingkronisasi dengan MySQL, seperti C<, C++, C#, bahasa pemrograman eiffel, bahasa pemrograman *Smaltalk*, bahasa pemrograman *java*, bahasa pemrograman *lisp*, *perl*, PHP, bahasa pemrograman *phiton*, *ruby*, *REALbasic*, dan Tcl. (Adelheid, 2012).

*MySQL* adalah *Relational Database Management* (*RDBMS*) yang didistribusikan secara gratis dibawah lisensi *General Public License* (*GPL*). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.Saat ini *MySQL* merupakan *software databas*e yang memegang kendali terbesar pada aplikasi *database* berbasis *web*. Arsitektur *database*nya sangat cocok bekerja pada lingkungan tersebut, didukung kemampuannya dalam berinteraksi dengan berbagai bahasa *programmer*an *side-server* seperti *PHP*, *ASP*, *CGI* dan *JSP* (Prasetyo, 2003).