BAB II

DASAR TEORI

A. DEFINISI GAJI

Menurut Soemarso S.R (2005:307) "Gaji adalah imbalan kepada pegawai yang diberi tugas-tugas administratif dan pimpinan yang jumlahnya biasanya tetap secara bulanan atau tahunan."

Gaji merupakan suatu bentuk balas jasa ataupun penghargaan yang diberikan secara teratur kepada seorang pegawai atas jasa dan hasil kerjanya. Gaji sering juga disebut sebagai upah, dimana keduanya merupakan suatu bentuk kompensasi, yakni imbalan jasa yang diberikan secara teratur atas prestasi kerja yang diberikan kepada seorang pegawai. Perbedaan gaji dan upah hanya terletak pada kuatnya ikatan kerja dan jangka waktu penerimaannya.. Dilihat dari jangka waktu penerimaannya, gaji pada umumnya diberikan pada setiap akhir bulan, sedang upah diberikan pada setiap hari ataupun setiap minggu.

B. PENGERTIAN DOSEN

1. Dosen

   Dosen adalah seorang yang berdasarkan pendidikan dan keahliannya diangkat oleh penyelenggara perguruan tinggi dengan tugas utama mengajar pada perguruan tinggi yang bersangkutan.[12]

2. Dosen Biasa

Dosen biasa adalah dosen yang diangkat dan ditempatkan sebagai tenaga tetap pada perguruan tinggi yang bersangkutan.[12]

3. Dosen Luar Biasa

Dosen luar biasa adalah dosen yang bukan tenaga tetap pada perguruan tinggi yang bersangkutan.[12]

C. PENGERTIAN APLIKASI

Apikasi merupakan penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses masukan (*input*) menjadi keluaran (*output*).[6] Suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna digunakan berbagai aplikasi. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Sebuah sistem komputer berpikir diatur oleh program. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan kedalam bahasa – bahasa tertentu sesuai dengan format yang ada pada *instruction set*. Program aplikasi merupakan program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh – contoh aplikasi ialah program pemproses kata dan *web browser*.

D. DEFINISI APLIKASI PENGGAJIAN

Penggajian merupakan suatu hal yang penting dilakukan pada setiap instansi atau perusahaan untuk meningkatkan semangat bagi para karyawannya agar terus berprestasi karena dengan gaji yang diperoleh seseorang dapat memenuhi kebutuhan hidupnya.

Aplikasi Penggajian merupakan suatu aplikasi yang dapat mengolah dan menyajikan data penggajian yang akan diterima oleh masing-masing pegawainya pada suatu instansi atau perusahaan. Karena itu program aplikasi penggajian juga sangat dibutuhkan. Dalam penggajian yang baik didapat jika didalamnya terdapat unsur-unsur program aplikasi penggajian seperti: menghitung gaji karyawan dengan lebih mudah serta meningkatkan kecepatan dan ketepatan dalam penyampaian informasi secara akurat. Suatu program aplikasi penggajian yang baik adalah jika didalamnya terdapat unsur-unsur program aplikasi penggajian seperti[1] :

1. Penyimpanan data

   Suatu kegiatan untuk memelihara dan menyimpan data. Semua data harus disimpan disuatu tempat sampai data diperlukan. Data tersebut disimpan dalam berbagai media penyimpanan, dan file yang disimpan disebut *database*.

2. Pengolahan data

   Proses operasi sistematis terhadap data yang dibutuhkan, yang didalamnya terdapat *input*, proses, dan *output*.

3. Penyajian data

Hasil dari proses pengolahan data yang akan disampaikan kepada para pegawai atau karyawan.

4. Pengontrolan data

   Suatu aktivitas untuk menjamin bahwa program aplikasi tersebut berjalan sesuai dengan apa yang diharapkan.

E. REKAYASA PERANGKAT LUNAK[13]

Rekayasa Perangkat Lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang *reliable* dan bekerja secara efisien pada mesin nyata. Sedangkan IEEE [IEE93] mengembangkan definisi yang lebih komprehensif mengenai rekayasa perangkat lunak,yaitu :

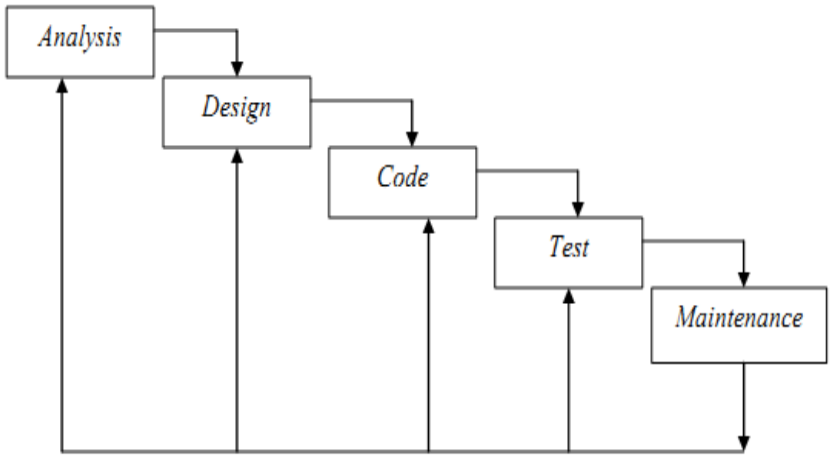
1. Aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak.
2. Studi tentang pendekatan-pendekatan seperti pada nomor satu.

F. REKAYASA SISTEM BERORIENTASI OBJEK[8]

Rekayasa adalah bentuk upaya manusia untuk mencari atau merancang produk yang berkualitas. Produk dapat berupa sistem atau perangkat lunak yang sedang atau akan dibangun dengan suatu model pengembangan. Orientasi objek adalah cara pandang bukan sekedar algoritma yang diimplementasikan pada bahasa pemrograman berorientasi objek, cara berpikir orientasi objek adalah

segala sesuatu dipandang sebagai objek. Objek dapat berupa konsep, abstraksi atau sesuatu dengan batas-batas tegas dan mempunyai arti untuk persoalan yang ditangani. Objek mempunyai identitas dan dapat dibedakan. Ada 2 (dua) kegunaan objek, yaitu untuk meningkatkan pemahaman dunia nyata dan menyediakan dasar pengetahuan praktis untuk implementasi komputer.

Model yang digunakan dalam proses perancangan perangkat lunak masih seperti model yang digunakan dalam metodologi konvensional, yaitu dapat menggunakan model *life cycle* atau *waterfall*. Dalam model *waterfall*, terdapat lima tahapan proses, yaitu tahap rekayasa sistem, analisis, perancangan, pemrograman, pengujian, pemeliharaan. Dengan menggunakan *waterfall*, memungkinkan adanya pengembangan aplikasi di kemudian hari, sehingga *software* yang dibangun dapat bertahan dalam jangka waktu yang panjang.



Gambar II.1 Tahapan Model *Waterfall*

1. Analisis (*Analysis)*

   Menentukan kegiatan dari rekayasa sistem dapat diimplementasikan menjadi sebuah sistem atau tidak dan menentukan prosedur-prosedur yang bekerja. Adapun prosedur-prosedur tersebut meliputi masukan (*input*), pemrosesan (*processing*), dan keluaran (*output*).

2. Perancangan (*Design*)

   Tahap merancang objek menggunakan desain berorientasi objek, secara spesifik mempunyai arti bahwa dalam mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Konsep berorientasi objek dapat dilakukan untuk semua sistem bahasa, dalam tugas akhir ini adapun yang digunakan yaitu UML (*Unified Modelling Language*) dan menggunakan beberapa diagram dari UML antara lain : *use case, diagram activity* dan *class diagram.*

3. Pengkodean (*Code*)

   Kegiatan yang mengimplementasikan hasil dari perancangan perangkat lunak ke dalam kode program agar pengguna (*user*) dapat memahami sistem yang sedang dibangun.

4. Pengujian (*Test*)

   Memfokuskan pada fungsi internal, fungsi eksternal dari perangkat lunak dan mencari segala kemungkinan kesalahan, memeriksa input data sesuai dengan hasil yang diinginkan setelah proses.

5. Pemeliharaan (*Maintenance*)

Kegiatan untuk memelihara (*maintenance*) perangkat lunak yang telah dibangun, pemeliharaan tersebut dilakukan agar keutuhan program dapat terjaga.

G. *UNIFIED MODELING LANGUAGE* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.[5]

UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan oleh XML (*Extensible Markup Language*) yaitu bahasa tambahan untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam.[16] Standar UML dikelola oleh OMG (*Object Management Group*).

Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML diantaranya: Benda (things), Hubungan *(Relationship)* dan Diagram.

1. Benda (*things)*

   Benda merupakan hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari sebuah konsep dan atau fisik.

2. Hubungan (Relationship)

   Hubungan sebagai alat komunikasi dari benda-benda. Ada 4 (empat) macam hubungan didalam penggunaan UML, yaitu:

   a. *Dependency* adalah hubungan semantik antara dua benda (*things*) yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah dependency digambarkan sebuah panah dengan garis terputus-putus.

   b. *Association* adalah hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *association* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.

   c. *Generalizations* adalah menggambarkan hubungan khusus dalam obyek anak (*child*) yang menggantikan obyek parent atau induk. Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah.

d. *Realizations* merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau element, serta antara *use cases dan collaborations*.

3. Diagram – Diagram UML[2]

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; *use-case, class, object, state, sequence, collaboration, activity, component,* dan *deployment diagram.*

a. Diagram *Use Case*

Diagram Use Case menggambarkan aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram Use Case dekat kaitannya dengan kejadian-kejadian. Kejadian (*scenario*) merupakan contoh yang terjadi ketika seseorang berinteraksi dengan sistem.

b. Diagram *Class*

Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis; menggambarkan hubungan yang terjadi masing-masing kelas.

c. Diagram *Sequence*

Diagram sequence merupakan salah satu diagram *Interaction* yang menjelaskan suatu operasi dilakukan: pesan yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

d. Diagram *Collaboration*

Diagram *Collaboration* juga merupakan diagram *Interaction*. Diagram ini membawa informasi yang sama dengan diagram *Sequence*, tetapi lebih memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan.

e. Diagram *StateChart*

Behaviors dan *State* dimiliki oleh obyek. Keadaan dari suatu obyek bergantung pada kegiatan dan keadaan yang berlaku pada saat itu. Diagram StateChart menunjukan kemungkinan dari keadaan obyek dan proses yang menyebabkan perubahan pada keadaannya.

f. Diagram *Activity*

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi dan terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan aktifitas-aktifitas tersebut bergantung satu sama lain.

g. Diagram Component dan Deployment

Component adalah sebuah kode-kode *module*. Diagram *Component* merupakan fisik sebenarnya dari diagram *Class*. Diagram *Deployment* menerangkan bahwa konfigurasi fisik software dan hardware.

H. *BORLAND DELPHI 7.0*

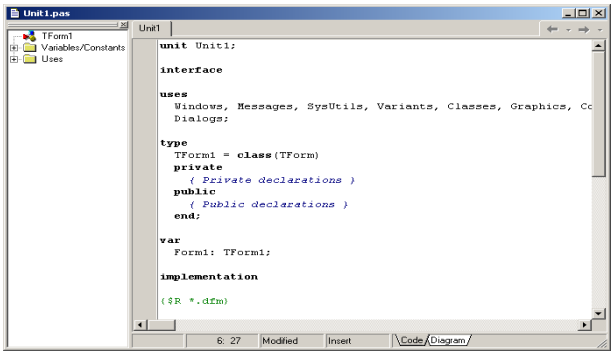
Delphi merupakan lingkungan pemrograman yang terintegrasi IDE (*Integrate Development Environment*). Delphi bukan bahasa pemrograman, tetapi perangkat lunak yang menyediakan seperangkat alat untuk membantu pemrograman dalam menulis program komputer. Delphi menggunakan *object pascal* sebagai bahasa pemrogramannya. *Object Pascal* merupakan bahasa *pascal* yang diberi tambahan kemampuan untuk menerapkan konsep-konsep OOP (*Object Oriented Programming*). Seluruh sintak *object pascal* menggunakan aturan yang ada di dalam *pascal*, termasuk perintah-perintah dasar seperti *control structures*, *variabels*, *array*, dan sebagainya.

Peralatan yang disediakan oleh Delphi memberikan kemudahan bagi pemrogram untuk membuat program secara *visual programming* yaitu metoda dimana sebagian atau keseluruhan program dibuat dengan cara menggambarkan tampilan atau hasil akhir dan kemudian meminta beberapa perangkat untuk membuat kode-kode program berdasarkan gambaran hasil akhir tersebut.[11]

Adapun beberapa peralatan yang disediakan oleh Delphi dan cukup diketahui antara lain:

a. Code Editor

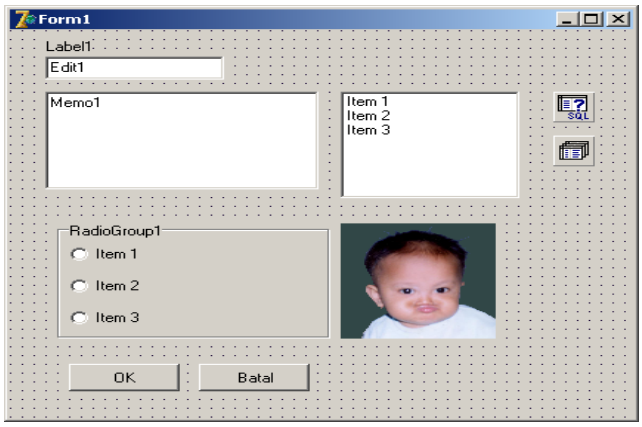
Code Editor merupakan peralatan yang digunakan untuk menuliskan kode-kode program. *Code Editor* menyediakan sejumlah fasilitas penyuntingan (editing) antara lain: *copy*, *cut*, *paste*, *find*, *replace*, dan sebagainya. *Code Editor* mengetahui penulisan merupakan perintah *Object Pascal* atau bukan dan menampilkan tulisan sesuai dengan tipe atau kelompok tulisan tersebut.



Gambar II.2. *Code Editor*

b. Form

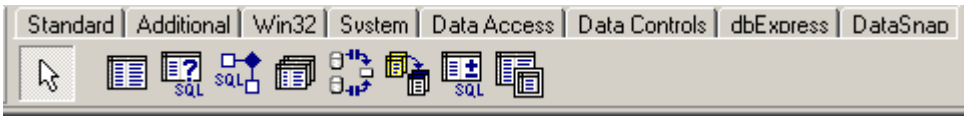
Form merupakan area pemrogram meletakkan komponen-komponen *input* dan *output*. Delphi akan secara otomatis membuat kode-kode program untuk membuat dan mengatur komponen-komponen tersebut. Umumnya pada setiap aplikasi ada paling tidak satu buah *form* dan *form* tersebut dijadikan sebagai form utama (*main form*).



Gambar II.3. *Form* dan Komponennya

c. Component Pallete

Component Pallete adalah peralatan yang menyediakan daftar komponen yang dapat digunakan oleh pemrogram.



Gambar II.4. *Compenent Pallete*

Komponen di dalam Delphi dibedakan menjadi dua macam, yaitu komponen visual dan non visual. Komponen Visual adalah komponen yang memberikan tampilan tertentu pada saat dimasukkan ke dalam form sedangkan komponen non-visual adalah komponen yang tidak memberikan tampilan tertentu saat dimasukkan ke dalam form. Komponen non-visual yang dimasukkan ke dalam form hanya ditampilkan sebagai sebuah kotak berisi simbol tertentu.

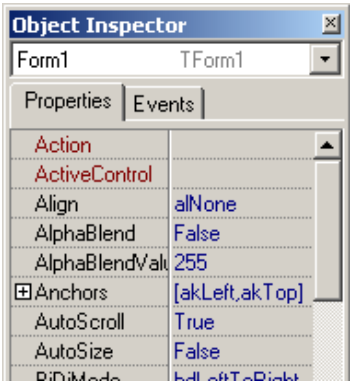
Gambar II.5. Komponen Visual

Gambar II.6. Komponen Non Visual

d. Object Inspector

Object Inspector adalah peralatan yang digunakan untuk mengatur properti dari komponen yang ada di *form* termasuk properti *form*. *Object Inspector* memberi dua macam peralatan, yaitu *properties* dan *event*.

Peralatan *properties* adalah peralatan yang digunakan untuk mengubah atau mengatur nilai-nilai dari properti komponen sedangkan peralatan *events* digunakan untuk membuat *event-handler*. *Event handler* adalah prosedur yang digunakan khusus untuk menanggapi satu *event* atau *message* tertentu.

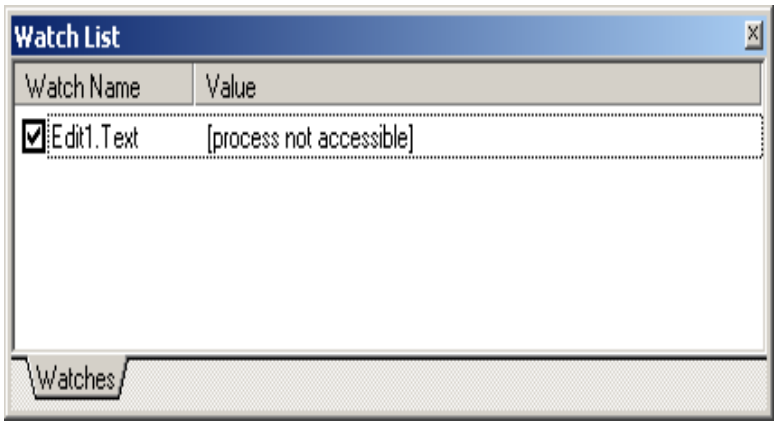


Gambar II.7. *Object Inspector*

e. Watch List

Watch List merupakan peralatan yang digunakan untuk memeriksa isi satu variabel atau properti tertentu saat *program* sedang dieksekusi. *Watch*

List biasanya digunakan bersamaan dengan *Break Points* dan *Step-by-Step execution*. Disamping itu juga dapat digunakan untuk mencari kesalahan di dalam *program* dengan cepat.[11]



Gambar II.8. *Watch List*

I. *MySQL*[3]

MySQL merupakan sistem manajemen basisi data SQL yang sangat terkenal dan bersifat *open source.* MySQL dibangun, didistribusikan dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang MySQL.

MySQL termasuk jenis RDMS (*Relational Database Management System*). Pada MySQL, sebuah *database* terdiri dari tabel – tabel. Sebuah tabel terdiri dari baris dan kolom.

Adapun fitur utama dari MySQL adalah:

a. Ditulis dalam bahasa C dan C++.

b. Bekerja dalam berbagai *platform*, misalnya Windows, Unix, Mac OS X, Solaris, dan lain-lain.

c. Menyediakan mesin penyimpanan (*engine storage*) transaksi dan non-transaksi.

d. *Server* tersedia sebagai program yang terpisah untuk digunakan pada lingkungan jaringan *client-server*.

e. Mempunyai sistem *password* yang fleksibel dan aman.

f. Dapat memanggil basis data dalam skala besar. Basis data dalam MySQL dapat berisi 50 juta *record*.

g. *Client* dapat terkoneksi ke MySQL menggunakan *socket* TCP/IP pada *platform* manapun.

J. JARINGAN KOMPUTER[9]

1. Pengertian

Jaringan komputer adalah himpunan "interaksi" antara dua komputer *autonomous* atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Bila sebuah komputer dapat membuat komputer lainnya *restart*, *shutdown*, atau melakukan kontrol lainnya maka komputer-komputer tersebut tidak *autonomous* (tidak melakukan kontrol terhadap komputer lain dengan akses penuh).

2. Tujuan Jaringan

Tujuan dari jaringan komputer adalah:

a. Komunikasi : contohnya surat elektronik, *instant messaging* dan *chatting*.
b. Akses Informasi : contohnya *web browsing*
c. Membagi sumber daya : contohnya berbagi pemakaian *printer*¸ CPU, memori dan *harddisk*.

3. Perangkat Jaringan Komputer

a. *Server*

Komputer *server* biasanya mempunyai sistem operasi dan *database* yang menyediakan layanan kepada komputer lain dalam jaringan. Jika jaringan dihubungkan ke dalam internet, komputer *server* juga berfungsi sebagai *gateway* atau gerbang komputer *client* untuk mengakses internet.

b. *Client*

Komputer *client* adalah komputer yang digunakan untuk melakukan pengolahan data yang diambil dari *server*.

c. Kartu Jaringan

Kartu jaringan atau LAN *Card* merupakan perangkat paling utama yang harus terpasang pada komputer agar komputer bisa terhubung dengan jaringan. Setiap komputer dapat dihubungkan dengan suatu jaringan melalui kartu jaringan.

d. *Switch / Hub*

Switch / Hub merupakan alat yang mempunyai fungsi sebagai tempat untuk menerima *file – file* data dari komputer untuk kemudian meneruskannya ke komputer lain pada suatu jaringan.

e. Kabel dan Konektor

Kabel dan Konektor adalah kabel jaringan yang digunakan untuk menghubungkan satu komputer dengan komputer lain. Kabel yang digunakan dalam jaringan adalah kabel *UTP*.

4. Klarifikasi Jaringan

Berdasarkan peranan dan hubungan tiap komputer dalam memproses data, jaringan komputer terdiri dari:

1. *Client – Server*

Jaringan komputer dengan komputer yang didedikasikan khusus sebagai *server*. Pada jaringan ini terdapat satu komputer *server* dan beberapa komputer *client*. Komputer *client* sebagai perantara untuk dapat mengakses data pada komputer *server* sedangkan komputer *server* menyediakan informasi yang diperlukan oleh komputer *client.*

a. Kelebihan dari tipe jaringan *Client – Server* sebagai berikut:

1) Teknologi baru dengan mudah terintegrasi ke dalam sistem.
2) Sistem administrasi dan keamanan jaingan lebih baik karena terdapat sebuah *node* yang bertugas sebagai *administrator* pada pada jaringan yang mengelola administrasi dan sistem keamanan pada jaringan.
3) Kecepatan *transfer* data lebih tinggi karena penyediaan fasilitas jaringan serta pengelolaannya dilakukan secara khusus oleh satu

komputer *server* yang tidak dibebani dengan tugas lain seperti *workstation*.

4) Sistem *back-up* data lebih baik karena pada jaringan *client – server bac-kup* dilakukan terpusat di *server,* yang akan mem*back-up* seluruh data yang digunakan di dalam jaringan.

b. Kekurangan dari tipe jaringan *Client – Server* sebagai berikut:

1) Adanya satu komputer khusus yang memiliki kemampuan lebih sebagai *server*.

2) Apabila *server* mengalami gangguan maka seluruh jaringan akan terganggu. Dengan kata lain keberlangsungan jalannya jaringan sangat bergantung pada *server.*

3) Biaya pengerjaan dan pengadaan bahan lebih mahal.

2. Peer to Peer

Apabila dilihat dari tugas *server* antara *client – server* dengan *peer to peer,* maka *server* di tipe *peer to peer* di istilahkan *non dedicated server* karena *server* berperan tidak sebagai *server* murni melainkan dapat pula berperan sebagai *worksation*.

a. Kelebihan dari tipe jaringan *Peer to Peer* sebagai berikut:

1) Komputer satu dengan komputer lain dalam jaringan dapat saling berbagi dalam pemakaian fasilitas yang dimilikinya seperti *drive, harddisk, printer* dan lainnya.

2) Biaya pengerjaan dan pengadaan bahan relatif lebih murah apabila dibandingkan dengan tipe jaringan *client – server,* karena tidak memerlukan adanya *server* yang memiliki spesifikasi khusus untuk mengorganisasikan fasilitas jaringan.
3) Apabila salah satu komputer mati atau rusak maka jaringan secara keseluruhan tidak akan mengalami gangguan. Dengan kata lain kerja jaringan tidak bergantung pada *server*.

b. Kekurangan dari tipe jaringan *Peer to Peer* sebagai berikut

1) Permasalahan pada jaringan relatif lebih rumit, karena tipe *peer to peer* setiap komputer memungkinkan terlibat dalam komunikasi yang ada dalam jaringan. Dalam jaringan *client – server* komunikasi hanya antara *server* dengan workstation.
2) Sistem keamanan pada jaringan ditentukan oleh masing - masing pemakai (*user*).
3) Dikarenakan data pada jaringan tersebar di setiap komputer dalam jaringan, maka *backup* harus dilakukan oleh masing - masing komputer tersebut.