BAB II

LANDASAN TEORI

2.1 Sejarah *Worm* Komputer

Gagasan *worm* dan virus berawal pada tahun 1949, saat seorang *founder* Electronic Discrete Automatic Computer (EDVAC), John Von Newman, memaparkan dalam sebuah papernya yang berjudul "Theory and Organization of Complicated Automata", dibahas suatu kemungkinan bahwa suatu program dapat melakukan penyebaran dengan sendirinya.

Kemudian pada tahun 1960-an, para peneliti AT&T Bell Laboratory membuat semacam permainan dengan menciptakan suatu program yang dapat memusnakan program ciptaan lawan, dan mampu bertahan terhadap serangan program lawan lainnya.

Pada akhirnya si-pemenang adalah pemilik program yang tersisa paling banyak. Para peneliti sadar akan bahaya program tersebut, maka setiaop selesai permainan program tersebut selalu dimusnakan.

Cikal bakal program *worm* pertama kali dibuat oleh Bob Thomas pada tahun 1971. Program ini merupakan solusi dari kebutuhan sistem kendali lalu lintas udara.

Program ini akan membantu mengingatkan operator apabila pengendali suatu pesawat udara berpindah dari satu komputer ke komputer lainnya. Sesungguhnya program yang disebut "*creeper*" ini hanya berpindah dari layar ke layar pada jaringan, dengan menampilkan pesan "*I'm creeper! Catch me if you*

can!". Tetapi program *creeper* tidak mereproduksi dirinya sendiri, hingga beberapa programmer lain mencoba membuat program serupa, tetapi gagasan tersebut berangsur-angsur hilang dalam beberapa bulan kemudian.

Istilah "virus" mungkin sudah tidak asing lagi terdengar, dapat dikatakan hampir setiap orang yang megenal komputer juga mengenal istilah ini, sementara istilah "*worm*" tidak begitu dikenal, padahal istilah ini diciptakan oleh John Shoch dan Jon Hupp di Xerox PARC (palo Alto Research Centre) pda tahun 1979 sebelum istilah virus komputer dipublikasikan.

Istilah *worm* ini ternyata diilhami oleh suatu program "tape*worm*" (cacing pita) dalam sebuah novel fiksi ilmiah karangan John Brunner yang berjudul "The Shockwave Rider", yang mengisahkan suatu pemerintahan totaliter yang mengendalikan warga negaranya melalui suatu jaringan komputer.

Untuk memerangi hal itu akhirnya seorang pahlawan dalam novel tersebut kemudian memenuhi jaringan dengan suatu program yang disebut "tape*worm*" sehingga memaksa mematikan jaringan komputer yang secara otomatis juga menghilangkan kendali pemerintah terhadap warga negaranya.

Kemudian diawal tahun 1980 John Shoch dan Jon Hupp mengadakan sebuah penelitian dengan mengembangkan lima buah program *worm*, masing-masing *worm* dibuat dengan tujuan tertentu yang membantu jaringan disekitarnya.

Beberapa *worm* terlihat sederhana, seperti *worm* "town crier" yang bertugas memasuki jaringan hanya untuk menampilkan pengumuman. *Worm* lainnya terlihat lebih kompleks dan pintar, seperti "vampire".

Worm ini tidak akan melakukan kegiatan pada siang harinya, tetapi saat malam hari *worm* akan melakukan suatu kegiatan tertentu yang telah terprogram sebelumnya, ini berguna untuk memanfaatkan komputer yang tidak bekerja pada malam hari dengan memebrikan tugas yang kompleks dan memerlukan daya proses yang lebih. Saat fajar, *worm* akan menghentikan pekerjaannya dengan terlebih dahulu menyimpan seluruh pekerjaan yang dilakukannnya malam itu dan menunggu sore berikutnya.

Walau bagaimanapun, walaupun program ini berguna tapi disadari bahwa program ini juga akan sangat berbahaya apabila asalah digunakan. Hal ini terbukti saat sebuah *worm* mengalami malfungsi pada suatu malam dan keesokan harinya para pekerja menemukan seluruh komputer yang ada pada jaringan tersebut mengalami *crash* (suatu kerusakan dimana sistem output dan input tidak berfungsi).

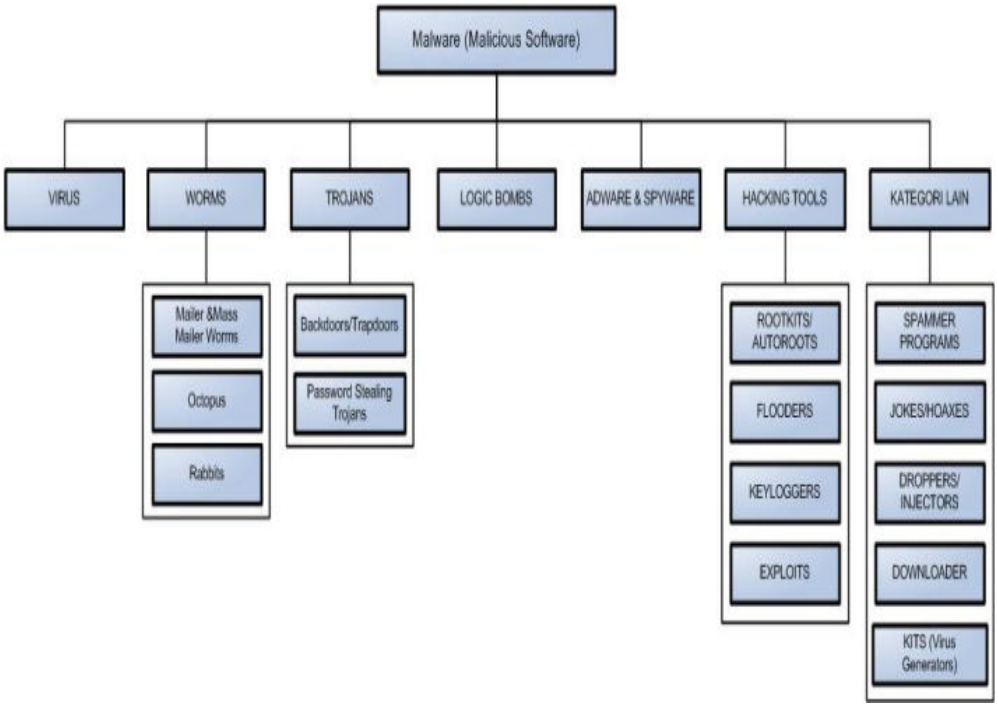
Lebih dari itu saat komputer dihidupkan kembali, *worm* malfungsi tersebut kembali membuat komputer *crash*. Pada akhirnya dibuatsemacam vaksin untuk mencegah *worm* tersebut, dan mulai saat itu penelitian terhadap *worm* tersebut ditutupu umum.

2.2 Perbedaan *Worm* dan Virus

Perkembangan teknologi komputer yang pesat, ternyata tidak hanya membawa manfaat yang besar bagi penggunanya tetapi juga menimbulkan dampak negatif dengan dibuatnya kode program yang berbahaya. Program-program berbahaya itu sering disebut juga dengan *Malicious Code/Malicious Software (malware)*. Pada dasarnya malware didefinisikan dengan sebuah

perangkat lunak yang didesain untuk melakukan infiltrasi atau merusak suatu sistem komputer, tanpa seijin pemiliknya.

Beberapa usaha untuk membuat pengklasifikasian malware secara tepat tidak mudah dilakukan, tumpang tindih antara spesifikasi teknis tiap kelas dan sub kelasnya selalu ditemukan pada klasifikasi tersebut. Namun secara umum, menurut Peter Szor malware dapat diklasifikasikan seperti gambar berikut ini:



Gambar 2.1 Klasifikasi Malware

Dari klasifikasi di atas, karena memiliki aksi yang hampir serupa, terdapat dua tipe malware yang agak sulit dibedakan yaitu, virus dan *worm*s. Untuk mempermudah melihat perbedaan kedua malware tersebut dapat ditinjau dari:

2.2.1 Definisi

Berikut akan dijelaskan mengenai virus dan worm.

2.2.1.1 Virus

Pada awalnya virus didefinisikan oleh Frederick B. Cohen sebagai suatu program yang dapat menginfeksi program lain dengan memodifikasinya, termasuk kemungkinan untuk berevolusi dengan menggandakan dirinya sendiri. Seiring dengan perkembangan teknik pemrogramannya, terdapat beberapa bentuk virus yang tidak sesuai dengan definisi tersebut. Sebagai contoh, suatu virus yang sering disebut companion virus memiliki kemampuan menggandakan diri tanpa mengubah program yang diinfeksinya. Sehingga menurut Peter Szor, definisi yang lebih akurat untuk virus pada saat ini adalah, suatu program yang secara berulang (recursively) dan dengan tegas (explicitly) menggandakan suatu versi dirinya sebagai kemungkinan untuk berevolusi.

2.2.1.2 *Worm*

Sedangkan definisi formal untuk *worm* menurut Robert T. Morris adalah suatu program yang berpindah dari satu komputer ke komputer yang lain tanpa mengikatkan dirinya (*attach itselft*) pada sistem operasi komputer yang diinfeksinya. Sejalan dengan perkembangannya, definsi *worm* tersebut sudah tidak begitu tepat. Beberapa *worm* sering menggunakan teknik untuk menyembunyikan kehadirannya dengan melakukan instalasi atau memodifikasi sistem. Sehingga definisi yang lebih tepat menurut Jose Nazario adalah suatu agen penginfeksi yang otonom dan independen dalam bereplikasi, serta memiliki kemampuan dalam menginfeksi sistem host baru melalui fasilitas jaringan.

2.2.2 Cara Penyebaran

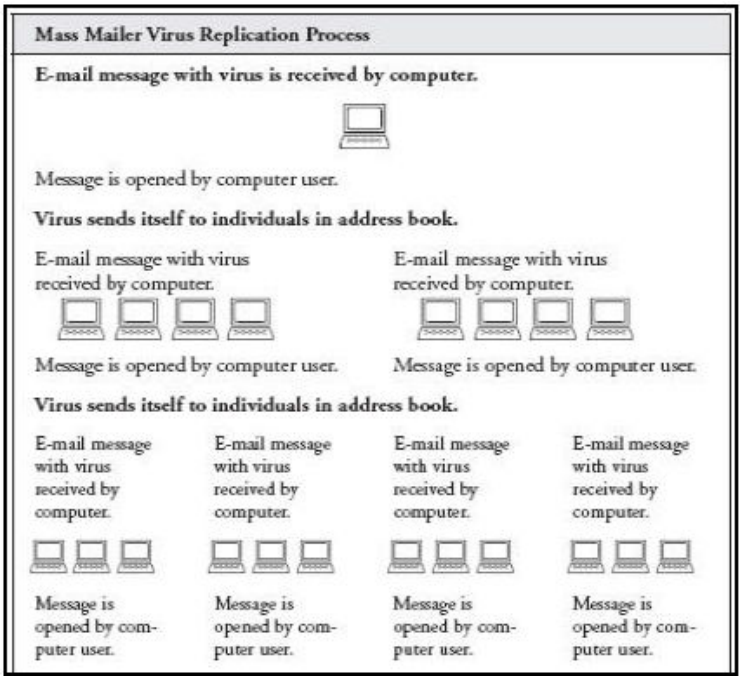
2.2.2.1 Virus

Virus memerlukan campur tangan pengguna dalam penyebarannya, misalnya dalam proses download, klik ganda pada file yang terinfeksi, dan lain-lain.

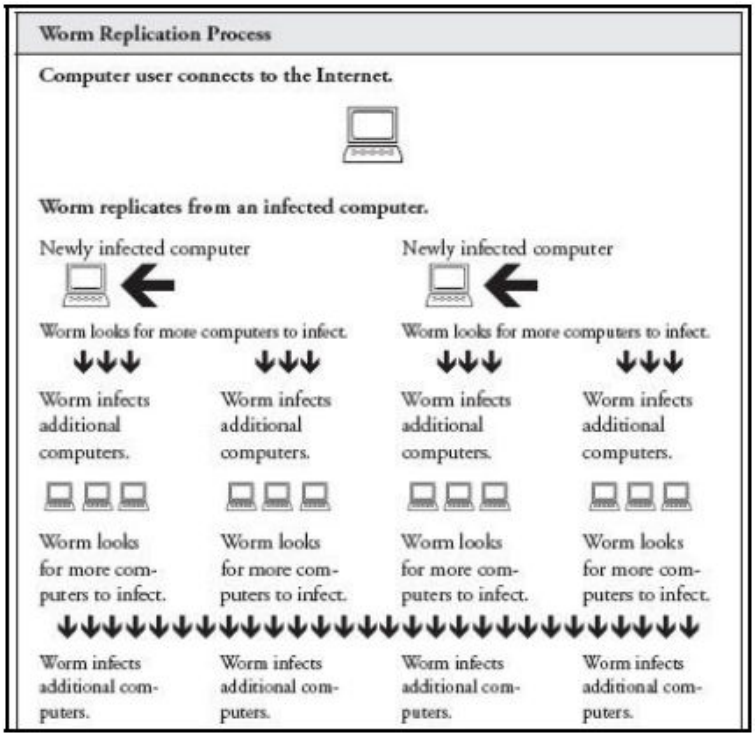
2.2.2.2 *Worm*

Sedangkan *worm* dapat secara otomatis menyebar dengan tanpa atau sedikit campur tangan dari penggunannya. Misalnya dengan satu kali klik pada file lampiran e-mail yang terinfeksi *worm*, maka satu atau beberapa sistem yang terkoneksi melalui e-mail tersebut akan segera terinfeksi.

Untuk lebih jelas, dapat dilihat dari dua contoh gambar berikut:



Gambar 2.2 Cara Penyebaran Virus



Gambar 2.3 Cara Penyebaran *Worm*

Istilah "virus" selalu digunakan sebagai suatu acuan umum untukn setiap malcode (program atau script yang dibuat dengan tujuan membahayakan atau merugikan sebuah sistem komputer), seperti *worm*, trojan bahkan hoax yang sesungguhnya bukan sebuah virus komputer, berikut adalah beberapa jenis malcode tersebut:

1. Virus Komputer: merujuk pada program yang memiliki kemampuan untuk ber-reproduksi, menularin program lain dan menjadikan file-file program tertular sebagai file *infector*.
2. *Worm* Komputer: merujuk pada program independen yang memiliki kemampuan untuk ber-reproduksi, menulari sistem komputer dan walaupun

mampu untuk menularin program lain namun tidak bertujuan untuk menjadikan file tertular tersebut sebagai suatu file *infector*.

3. Trojan Horse: merujuk pada program independen yang tampaknya berguna, dan ketika dieksekusi, tanpa sepengetahuan pengguna, juga melaksanakan fungsi-fungsi yang bersifat destruktif dan merugikan.
4. Malicious Toolkits: merujuk pada program yang didesain untuk membantu menciptakan program-program yang dapat membahayakan sebuah sistem komputer. Contoh dari program jenis ini adalah toll pembuat virus dan program yang dibuat untuk membantun proses *cracking* atau *hacking*.

Dari beberapan keterangan di atas dapat diperjelas bahwa *worm* adalah suatu algoritma atau program yang mereproduksi diri sendiri dari sistem ke sistem dengan menggunakan media penyimpanan atau atau suatu jaringan.

Worm tidak menginfeksi file program lain dengan tujuan menjadikan file terinfeksi tersebut sebagai file *infector*. *Worm* mampu bekerja tanpa interaksi user, bisa merusak sebuah data secara langsung atau menurunkan kinerja sistem dengan "mengikat" sumber daya sistem komputer dan bahkan bisa mematikan sebuah jaringan. Berbeda dengan virus yang melakukan infeksi dengan "menumpang" pada file program lain, menunggu interaksi user dan menjadikan file terinfeksi sebagai file *infector*.

2.3 Kemampuan Dasar *Worm*

Worm umumnya memiliki kemampuan-kemampuan dasar berikut ini:

2.3.1 Kemampuan Reproduksi dan Distribusi

Yaitu kemampuan yang mutlak dimiliki suatu *worm* untuk membuat salinan dirinya, sekaligus mendistribusikan salinan tersebut pada sistem yang lain baik melalui media penyimpanan seperti disket, USB flash disk maupun melalui suatu jaringan komputer. Walaupun memiliki rutin untuk menginfeksi program lain namun tidak bertujuan menjadikan file program terinfeksi menjadi suatu file *infector*.

Pada awalnya *worm* dibuat dengan aksi memenuhi *harddisk* dan jaringan, namun seiring dengan perkembangan teknologi informasi hal ini akhirnya banyak ditinggalkan *worm* writer karena malah akan mengurangai kemampuannya untuk menyembunyikan diri, yang akan berakibat *worm* tersebut cepat "terendus" oleh *advanced user* atau bahkan perusahaan-perusahaan antivirus.

2.3.2 Kemampuan Rekayasa Sosial

Karena file *infector worm* akan aktif saat *user* mengeksekusinya maka *social engineering* atau rekayasa sosial menjadi hal yang sangat penting bagi suatu *worm*.

Layaknya seorang penjual yang mati-matian merayu calon pembeli maka *worm* akan "merias" programnya dengan icon dan nama yang sangat memikat agar *user* mengeksekusinya. Suatu *worm* bisa saja membuat salinan dirinya dengan nama file "porno" dan dengan gambar icon yang sangat tidak mencurigakan.

2.3.3 Kemampuan Menyembunyikan Diri

Menjaga tetap tidak diketahui adalah penting untuk *worm* jaman sekarang agar tetap bertahan pada suatu sistem. Hal ini biasanya dilakukan dengan cara tidak menampilkan sesuatu dari *worm* baik berupa suara maupun tampilan visual, menyembunyikan program *worm* dari *taskbar* bahkan dari jendela *tasklist*.

2.3.4 Kemampuan Mendapatkan Informasi

Suatu *worm* harus bisa mendapatkan informasi yang ia butuhkan, seperti jenis sistem operasi yang digunakan, direktori root, direktori sistem windows bahkan *worm* umumnya memerikasa suatu sistem apakah telah terpasang antivirus atau tidak, lebih jauh lagi *worm* akan berusaha mengenali jenis antivirus yang terpasang.

2.3.5 Kemampuan Mengadakan Manipulasi

Umumnya manipulasi dilakukan oleh *worm* untuk bertahan hidup. *Worm* cenderung mengadakan manipulasi pada registry agar *worm* bisa tetap aktif saat komputer dihidupkan, bahkan manipulasi registry milik suatu antivirus agar tidak mengganggu *worm* tersebut. Tapi *worm* bisa saja mengadakan manipulasi yang terlepas dari tujuan tadi, seperti mengubah volume label pada *harddisk* atau disket.

2.4 Siklus Hidup *Worm*

Siklus hidup *worm* secara umum melalui empat tahap, yaitu:

2.4.1 Propagation Phase (Fase Penyebaran)

Pada fase ini *worm* akan membuat salinan dirinya ke suatu tempat, baik pada media penyimpanan *fix disk* (tetap) atau *removable disk* (dapat dipindahkan), adapaun penyebarannya dapat dilakukan pda sistem lokal, jaringan atau internet.

2.4.2 Dormant Phase (Fase Istirahat/Tidur)

Pada fase ini *worm* tidaklah aktif. Wrom akan diaktifkan oleh suatu kondisi tertentu, semisal: tanggal yang ditentukan, kehadiran program lain/dieksekusinya program lain, dan sebagainya. Tidak semua *worm* melalui fase ini.

2.4.3 Trigerring Phase (Fase Aktif)

Di fase ini *worm* tersebut akan aktif dan menetap pada memori, hal ini dipicu oleh metode *launcher* yang digunakan *worm* tersebut.

2.4.4 Execution Phase (Fase Eksekusi)

Pada fase inilah *worm* yang telah aktif tadi akan melakukan fungsinya. Seperti menghapus file, menampilkan pesan-pesan dan sebagainya.

2.5 Metode File Name Scanning

Metode ini didasarkan pada file-file worm yang sudah diketahui sebelumnya. Data-data yang diperiksa dari trik tersebut disebut dengan istilah signature. Dalam metode ini sebuah *file name* (nama berkas) sangatlah penting diketahui dalam rangka mendeteksi suatu file yang berbahaya atau mengancam sistem komputer. Lebih jauh lagi dalam hal pencarian *file worm*, selain nama yang sudah diketahui diperlukan suatu validasi tambahan, yaitu *extension* dan besar kapasitas dari suatu file. Hal ini diperlukan agar metode tidak salah mendeteksi berkas-berkas karena

kemiripan yang dilihat dari nama file saja. Satu hal lagi, metode ini juga mendeteksi berkas-berkas yang sifatnya *hidden* (tersembunyi).

2.6 Metode String Scanning

Metode ini pada dasaranya sama dengan metode *file name scanning*. Hal yang perlu diketahui dalam metode ini adalah mendeteksi bukan berdasarkan nama berkas, tetapi berdasarkan string (dalam hal ini sederetan angka hexa). Dalam pengaplikasiannya, metode ini mengambil sample string hexa dengan bantuan tool yang bernama "hxd", sebuah tool yang dapat meng-*generate* atau melihat struktur bilangan hexa yang terkandung di dalam *file* atau berkas.

2.7 Perkembangan *Worm* Non Lokal

Berikut ini beberapa catatan singkat tentang *worm* yang pernah ada dan membuat banyak kerugian bagi para pengguna komputer.

2.7.1 Christma Exec

Pada tanggal 9 Desember 1987, *worm* "Christma Exec" menjadi *worm* pertama yang mampu menyebar dengan menggunakan media *e-mail* diantara komputer *mainframe* IBM. Wrom ini juga menjadi contoh penggunaan *social engineering*, dengan mengajak *user* untuk mengeksekusi *worm* tersebut dengan dalih akan menampilkan gambar pohon natal.

Worm tersebut memang menghasilkan gambar pohon natal pada layar monitor (digambar dengan menggunakan bahasa script yang disebut Rexx), tetapi *worm* tersebut juga mengirimkan salinan dirinya dengan menggunakan nama *user* kepada setiap nama yang ada pada daftar *e-mail* penerima, sehingga penerima

percaya bahwa *e-mail* yang dikirimkan tersebut adalah benar dari *user* yang dikenal dan bersedia membukanya.

2.7.2 Morris

Pada tanggal 2 Nopember 1988, *worm* Morris yang terkenal pada waktu itu berhasil melumpuhkan 6000 komputer dalam beberapa jam. *Worm* tersebut dibuat oleh seorang siswa Cornell, Robert Morris Jr. Kemudian diadakan penyelidikan, sampai akhirnya Morris dijatuhkan hukuman pada tahun 1990.

Kesimpulan yang diperoleh adalah motivasi dalam menulis *worm* tersebut tidak diketahui dan *worm* tidak diprogram untuk sengaja melakukan pengrusakan, tetapi kerusakan yang ditimbulkan disebabkan oleh kecelakaan dan kesalahan pemrograman.

Dibulan Oktober 1989, muncul sebuah *worm* bernama WANK (*Worm*s Against Nuclear Killers) yang tampaknya belajar dari *worm* Morris dan melakukan penularan pada komputer VMS pada DECNet. *Worm* ini menyebar dengan memanfaatkan *e-mail* dan mengekploitasi sistem untuk mendapat hak akses dengan berusaha mencari *account user name* dan *password*.

2.7.3 Happy99

Pada bulan Januari 1999, *worm* happy99 menyebar lewat *e-mail* dengan *attachment* sebuah *file* aplikasi bernama happy99.exe. Ketika *file* tersebut dieksekusi tampil gambar kembang api untuk memperingati tahun baru 1999, tetapi secara diam-diam memodifikasi file WSOCK32.DLL (file sistem untuk koneksi internet) dengan suatu program trojan horse yang mengijinkan *worm*

tersebut menyisipkan dirinya pada proses komunikasi internet, sementara *file* WSOCK32.DLL yang asli diubah kembali namanya menjadi WSOCK32.SKA.

2.7.4 Melisa

Di bulan Maret 1999, sebuah virus macro "Melisa" kembali meresahkan pengguna internet dengan menginfeksi 100.000 unit komputer hanya dalam waktu tiga hari.

Virus tersebut memulai penyebarannya dengan pengiriman perdana ke Usenet Newsgroup "alt.sex" yang menjanjikan *account name* berikut *password* untuk dapat mengakses sebuah situs erotis. Sebuah perusahaan antivirus Norton menyebutkan bahwa virus ini adalah penggabungan antara *worm* dan virus.

Melisa menyertakan sebuah file *attachment* berupa *file* dokumen *Word* yang terinfeksi. Ironisnya saat itu masih banyak yang percaya bahwa dengan membuka sebuah *e-mail* tidak dapat menginfeksi sebuah komputer.

Ketika macro tersebut tereksekusi oleh aplikasi *Word*. Pertama kali yang dilakukannya adalah melakukan pemeriksaan apakah versi aplikasi *Word* yang digunakan bisa diinfeksi, jika bisa maka virus akan mengurangi pengaturan keamanan pada aplikasi *Word* untuk mencegah aplikasi menampilkan pesan atau peringatan tentang adanya suatu macro, yang akan mencurigakan *user*.

Virus kemudian mencari sebuah *key* pada registry yang mengandung kata "kwyjibo", apabila *key* tersebut tidak ditemukan, virus akan mengeksekusi aplikasi Outlook dan berusaha mengirimkan salinan dirinya kepada 50 penerima yang ada pada *address book* (buku alamat) aplikasi Outlook. Sebagai tambahan virus menulari *file template* aplikasi *Word* "normal.dot" menggunakan fitur VBA

macro “Auto Execute”, *file* dokumen yang berasal dari *file template* tersebut akan membawa serta virus tersebut.

2.7.5 Pretty Park

Sebuah *worm* lainnya menyebar luas dimusim panas tahun 1999. *Worm* yang dinamakan “Pretty Park” ini menyertakan *attachment* berupa *file* “Pretty Park.exe”. tidak ada penjelasan pada *attachment*, hanya saja *file* tersebut menggunakan icon bergambar seekor beruang, yang merupakan sebuah karakter pada suatu pertunjukan televisi “South Park”.

Jika dieksekusi, *worm* menginstal dirinya ke direktori Sistem Windows dan memodifikasi registry yang membuat *worm* tersebut aktif saat *file* ber-ekstensi “exe” apa saja dieksekusi. Hal ini menjadi permasalahan bagi program antivirus, yang tentunya juga ber-ekstensi “exe”.

Worm juga mengirimkan salinan dirinya pada alamat *e-mail* yang ada pada buku alamat Windows. Pretty Park kemudian berusaha mengirimkan beberapa data sistem dan *password* pada sebuah server IRC (internet relay chat) tertentu, terakhir dilaporkan bahwa *worm* ini memasang suatu *backdoor*.

2.7.6 Explore Zip

Pada bulan Juni tahun 1999, muncul sebuah *worm* bernama “ExploreZip” yang menyamar sebagai file zip (file terkompresi) dalam *attachment* sebuah *e-mail* yang jika dieksekusi akan menampilkan pesan kesalahan. exploreZip secara diam-diam menyalin dirinya kedalam direktori sistem Windows dan memodifikasi registry.

Seperti *worm* lainnya, ExploreZip juga melakukan penyebaran lewat *e-mail* dengan memanfaatkan aplikasi Outlook atau Exchange, dengan mengawasi *e-mail* yang masuk dan membalas *e-mail* tersebut dengan salinan dirinya.

Pada tanggal 9 Januari 2003 ExploreZip kemudian dilaporkan menghasilkan varian baru. Varian ini bernama ExploreZip.N varian dan cukup merepotkan pengguna komputer di Indonesia. Salah satu metode penyebarannya menggunakan *e-mail* yang memiliki *attachment* berupa *file* ZIPPED_FILES.EXE saat *file* ini dieksekusi maka *worm* menginstal dirinya sendiri pada sistem Windows.

Worm ini meng-*overwrite* (menimpa) *file* dokumen dengan ekstensi DOC (Microsoft Word), XLS (Microsoft Excel), PPT (Microsoft PowerPoint), ASM (Assembler), CPP (C++), H (Header C) sehingga berkas-berkas tersebut sulit untuk diselamatkan.

2.7.7 Bubble Boy

Awal tahun 2000, muncul sebuah virus yang membawa sebuah konsep baru "BubbleBoy". Virus ini menunjukan bahwa sebuah komputer dapat tertular hanya dengan melihat *e-mail*, tanpa harus membuka pesannya.

Virus ini mengambil keuntungan dengan adanya selah keamanan pada aplikasi Internet Explorer, yang secara otomatis mengeksekusi script Visual Basic yang terdapat pada *body e-mail*. Virus akan datang sebagai sebuah *e-mail* dengan subjek "bubbleBoy is back", pesan berbentuk *file* HTML dan mengandung script virus dalam bahasa Visual Basic.

Jika menggunakan aplikasi Outlook, script tersebut akan dijalankan walaupun yang dilakukan hanya *preview*. *File* tersebut akan ditambahkan pada direktori

StartUp Windows, sehingga apabila komputer dihidupkan virus akan berusaha mengirimkan dirinya pada setiap alamat yang ada pada *address book* aplikasi Outlook. Diwaktu yang hampir bersamaan, *worm* "KAK" tersebar dengan cara yang serupa.

2.7.8 Love Letter

Dibulan Mei 2000, lajunya penyebaran *worm* "LoveLetter" menunjukkan efektifitas serangan dengan metode *social engineering*, yang hingga saat ini sudah menjadi suatu kebiasaan suatu *worm* dalam penyebarannya.

E-mail yang berisi *worm* ini memiliki subjek "I Love You" yang berarti "Saya Cinta Kamu" dan berisi pesan-pesan yang mendorong *user* untuk mengeksekusi *attachment* yang merupakan *worm* tersebut. File *attachment* berupa *Visual Basic Script* yang bisa dieksekusi dengan *Windows Script Host* (bagian dari Windows 98, Windows 2000, Internet Explorer 5, atau Outlook 5).

Ketika dieksekusi, LoveLetter menginstal dirinya ke dalam direktori Sistem Windows dan memodifikasi *registry* untuk memastikan bahwa *worm* akan aktif saat kompuer dihidupkan. Ketika komputer lainnya terinfeksi, dan jika aplikasi Outlook terinstal pada komputer tersebut, maka *worm* akan mengirimkan salinannya pada siapa saja yang ada pada *address book* aplikasi Outlook.

Sebagai tambahan *worm* akan membuat koneksi IRC dan mengirimkan salinan dirinya pada siapa saja yang brgabung pada saluran IRC tersebut. LoveLetter juga memilki kemampuan untuk mencuri *password*.

Dengan mengubah *home page* (alamat url yang akan diakses pertama kalinya) pada Internet Explorer ke suatu *website* di Asia, *worm* akan mengusahakan agar

suatu *trojan horse* di *download* dari *website* tersebut, dimana *trojan horse* tersebut akan mengumpulkan *password e-mail* dan mengirimkannya ke suatu alamat di Asia.

2.7.9 Hybris

Dibulan Oktober 2000, *worm* Hybris menyebar dengan *e-mail* ber-*attachment*. Jika dieksekusi akan memodifikasi file WSOCK32.DLL dalam rangka menjejaki semua lalu lintas ber-internet. Untuk setiap *e-mail* yang terkirim *worm* akan mengirimkan salinan dirinya ke alamat penerima yang sama. Yang menarik dari *worm* ini yaitu; bisa men-*download file update* untuk dirinya sendiri dari *newsgroup* "alt.comp.virus" metode yang digunakan termasuk canggih dan sangat berbahaya karena *payload worm* tersebut bisa diubah kapan saja.

2.7.10 Anna Kournikova

Pada bulan Februari 2001, kembali sebuah *worm* mencemaskan para pengguna internet, yang memanfaatkan kepopuleran seorang petenis asal Rusia "Anna Kournikova". *Worm* ini dibawa dalam suatu *attachment e-mail* yang menyatakan bahwa lampiran tersebut adalah *file* gambar dalam bentuk *file* jpg yang memuat foto pemain tenis tersebut. Apabila *file* tersebut dieksekusi maka akan mengirimkan salinan dirinya kepada setiap nama yang terdaftar pada buku alamat Microsoft Outlook.

2.7.11 Sadmind

Pada bulan Mei 2001, *worm* Sadmind menyebar dengan mentargetkan 2 *vulnerability* (kelemahan) pada 2 sistem operasi yang berbeda, dan menjadi teladan untuk *worm* berikutnya yang bisa melakukan serangan dengan berbagai

kombinasi. Pertama kali yang dilakukannya adalah meng-eksploitasi *vulnerability buffer overflow* pada sistem operasi Sun Solaris, dan menginstal suatu program agar bisa malakukan komunikasi dengan IIS *webserver* guna melakukan suatu serangan. *Vulnerability* ini kemudian diumumkan pada tanggal 18 Juni 2001, yang menunjuk sebagai *vulnerability* pada *Index Server* ISAPI.

2.7.12 Code Red

Memanfaatkan *vulnerability* pada *Index Server* ISAPI tersebut, pada tanggal 12 Juli 2001 muncul sebuah *worm* dengan nama "Code Red" yang menyerang semua IIS *webserver*, dengan aksi mengubah tampilan awal *website* pada *server* yang tertular.

Pertama kali *worm* menginstall dirinya pada sistem, membaca IP sistem dan dari IP tersebut *worm* menyusun 99 IP baru, kemudian melakukan pemeriksaan sistem operasi pada IP yang berhasil disusun. Jika *worm* menemukan sebuah IP target menggunakan sistem operasi Microsoft Windows maka *worm* akan meng-eksploitasi server target tersebut, dan melakukan *deface* (mengubah halaman awal suatu *website*) dengan tampilan "**Welcome to the www.*worm*.com ! hacked by Chinese!.**"

Berikutnya *worm* mencari *file* c:\not*worm*. *Worm* tidak akan menghentikan serangannya jika file tersebut tidak ditemukan. Pada tanggal 20 Juli *worm* akan melakukan serangan DOS (denial of service) pada server www.whitehouse.gov, kemudian pada tanggal 27 *worm* membuat dirinya dalam kondisi *dormant* (fase saat *worm* menjadi tidak aktif) secara permanent.

Tidak begitu lama pada tanggal 19 Juli 2001, muncul Code Red 1 yang merupakan versi kedua dari *worm* Code Red dengan penyebaran yang lebih cepat. Banyak perbaikan pada program *worm* sehingga mampu menginfeksi 359.000 unit komputer hanya dalam waktu 14 jam, seperti versi pertama *worm* ini juga membuat dirinya dalam kondisi dormant pada tanggal 20 Juli secara permanent.

Di bulan Agustus 2001, kembali Code Red muncul dengan versi berbeda "Code Red II", muncul dengan *payload* yang sangat berbahaya. *Worm* ini masih memanfaatkan *vulnerability* yang sama dengan versi sebelumnya, sedikit perubahan pada program, *worm* ini akan membuat suatu trojan "explorer.exe" dan ditempatkan pada direktori *root*.

2.7.13 Nimda

Tanggal 18 September 2001, *worm* Nimda adalah *worm* yang termasuk paling banyak menginfeksi komputer di Indonesia, muncul dengan memuat *payload* yang sangat berbahaya dan menggunakan beragam cara dalam penyebarannya, pada 12 jam pertama saja *worm* ini sudah berhasil menginfeksi 450.000 unit komputer, dan dalam dua hari Nimda berhasil menginfeksi 2,2 juta komputer serta menyebabkan kerugian sebesar US$370 juta.

Walaupun *worm* ini tidak menggunakan metode baru dalam penyebarannya, tetapi dengan penggabungan beberapa metode dalam suatu *worm*, menunjukan adanya tingkatan baru atas kompleksitas yang tidak terlihat sebelumnya. Nimda melakukan penyebaran dengan mengirimkan salinan dirinya melalui *e-mail* dengan subjek *random* (acak) dan sebuah file *attachment* "readme.exe".

Nimda memanfaatkan celah keamanan pada Internet Explorer dan Outlook Express 5.01 dan 5.5 Service Pack 1, dimana *e-mail* yang dikirimkan Nimda berupa halaman html yang memiliki *body script* tertentu dengan tujuan akan secara otomatis mengaktifkan file *attachment* walaupun *e-mail* tersebut hanya dilihat pada *preview pane* saja.

Bila komputer terinfeksi berada dalam suatu jaringan maka secara otomatis Nimda mencari direktori yang di *sharing* dari komputer lain dan memberikan hak tulis penuh (full access) kemudian membuat salinan dirinya ke dalam direktori-direktori tersebut, dengan mengambil nama file secara *random* pada daftar *file* yang ada pada komputer terinfeksi, dengan menggunkan eksensi EML atau NWS.

Pada server yang terinfeksi, *file-file* halaman web akan disisipkan *script* baru yang secara otomatis mengakses *file* readme.eml saat halaman web tersebut terbuka. Dalam penyebarannya, Nimda juga memanfaatkan *backdoor* pada server yang terinfeksi Code Red 2 atau Sadmind. Banyaknya perubahan pada *file* sistem dan *registry* membuat *worm* ini sulit untuk dibersihkan.

2.7.14 BadTrans.B

Pada bulan November 2001, BadTrans kembali beraksi setelah kemunculan perdananya pada tanggal 12 April 2001 dengan teknik yang lebih canggih dan bisa dikatakan sebagai era baru dalam *social engineering*.

BadTrans.B memeriksa direktori inbox pada aplikasi Outlook dengan mencari *e-mail* yang belum dibuka oleh *user*, kemudian membalas *e-mail* tersebut dengan *file attachment* berupa salinan dirinya sendiri.

Seperti Nimda, *e-mail* yang dikirimkan berupa halaman html dan secara otomatis mengaktifkan *file attachment* walaupun *e-mail* tersebut hanya dilihat pada *preview pane* saja. Pada proses pengiriman *e-mail* BadTrans.B menambahkan *underscore* (tanda garis bawah, "_") disetiap awal alamat pengirim, sehingga apabila *e-mail* tersebut di-*reply* (balas) maka *e-mail* balasan tersebut tidak akan sampai pada alamatnya.

BadTrans.B akan menginstal suatu program trojan yang akan mencuri *user name* dan *password* kemudian mengirimkannya pada suatu alamat *e-mail* tertentu, dilaporkan *worm* ini mengakibatkan kerugian sebesar US$210 juta.

2.7.15 Slammer

Serangan *worm* Slammer dimulai pada tanggal 23 Januari 2003, *worm* yang hanya berukuran 369 byte ini menjadikan Slammer sebagai *worm* dengan ukuran terkecil yang pernah ada. Cara kerja *worm* ini hampir sama dengan CodeRed, dan tidak memodifikasi *registry*.

Slammer memanfaatkan *vulnerability* yang ditemukan oleh Next Generation Security Software limited pada bulan juli 2002 dimana dengan memanfaatkan vulnerability ini, penyerang dapat menguasai SQL Server dan kode pemrograman yang brhasil dimasukan akan berjalan sebagai sistem karena hak dari MS SQL Serer di dalam komputer adalah sistem. *Worm* ini bukan merupakan *mass mailer* dan hanya menyebarkan dirinya melalui *scanning port* 1434.

2.7.16 Bagle

Di minggu ketiga Januari 2004, sebuah *worm* yang disinyalir berasal dari Jerman ternyata sukses meraih peringkat pertama sebagai *worm* yang paling

banyak dihentikan. *Worm* yang kemudian diketahui bernama Bagle ini memiliki siklus hidup yang tergolong singkat, Bagle memasuki fase *dormant* secara permanent pada tanggal 28 Januari 2004 dan berusaha untuk menghapus *file launcher*-nya.

Bagle memiliki ukuran *file* sebesar 15 KB, dan malakukan penyebaran dengan cara mengirimkan *file infector* melalui *e-mail* ber-*attachment*, dengan tujuan pengiriman yang dikoleksi dari *file-file* berekstensi txt, htm, dan wab pada sistem lokal.

Saat *file infector* dieksekusi, Bagle membuat salinan *file worm* pada direktori sistem Windows, kemudian memanipulasi *registry* agar *worm* tereksekusi setiap kali Windows *starup*.

2.7.17 Netsky

Tidak begitu lama setelah kemunculan *worm* Bagle, sebuah *worm* lainnya menyebar dengan kecepatan yang lebih tinggi. *Worm* Netsky melakukan penyebaran dengan cara yang sama seperti yang dilakukan *worm* Bagle, hanya saja Netsky memiliki beberapa rutin yang memungkinkan dirinya dapat melakukan penyebaran dengan menggunakan media aplikasi *peer to peer*.

Netsky mengirimkan salinan dirinya melalui sebuah *e-mail* ber-*attachment* dan dalam bentuk *file* terkompresi. *File worm* berukuran sebesar 22 KB dan menggunakan *icon* yang disamarkan sehingga terlihat sebagai sebuah *file* dokomen Microsoft Word.

Saat *file infector* dieksekusi Netsky membuat salinan *file worm* pada direktori Windows dengan nama *file* 'services.exe', kemudian memanipulasi beberapa nilai *registry* yang bertujuan agar *worm* dapat tereksekusi setiap kali Windows *startup*.

2.8 Perkembangan *Worm* Lokal

Berikut ini beberapa catatan singkat tenatang *worm* lokal yang pernah ada dan dibuat oleh *worm writer* asal Indonesia.

2.8.1 I-*Worm* Perkasa

Pada tanggal 6 Desember 2001 muncul sebuah *worm* lokal. *Worm* yang terkompresi dengan UPX ini memiliki ukuran sebesar 12,288 KB dan dibuat dengan menggunakan program Visual Basic.

Worm yang juga disebut dengan nama I-*Worm*.Imelda atau I-*Worm* Updater ini menggunakan media Microsoft Outlook dalam penyebarannya, saat *file infector* dieksekusi, *worm* ini meng-*copy* dirinya ke direktori Windows dengan nama 'UPDATE.EXE', kemudian menambahkan suatu nilai *registry* pada *sub key* 'Run' agar *worm* ini tetap aktif saat Windows *Startup*.

2.8.2 Pesin

Worm lokal lainnya muncul dipertengahan bulan September 2003 dengan menggunakan *icon* Microsoft Word. *Worm* yang terkompresi dengan ASPack ini juga sering dipanggil dengan sebutan Kenangan, Puisi Cinta, Halo, Mistery atau Myheart. Hal ini disebabkan karena *worm* Pesin menggunakan nama-nama tersebut sebagai nama dari *file infector*.

Worm Pesin menggunakan media disket dalam proses penyebaran, hal ini sangat efektif mengingat penggunaan disket masih sangat diminati oleh pengguna komputer di Indonesia.

Saat *file infector* dieksekusi, *worm* ini meng-*copy* dirinya ke direktori Windows dengan nama 'Systask.exe'. Pesin juga membuat salinan pada direktori My Documents dengan nama *file* 'MyHeart.exe' kemudian menambahkan suatu nilai *registry* pada *sub key* 'Run' agar *worm* ini tetap aktif saat Windows *startup*, yang menarik dari *worm* ini adalah *worm* akan menonaktifkan program Registry Editor jika *user* berusaha untuk menjalankannya.

2.8.3 Tabaru

Pada awal Januari 2005 kembali diketahui sebuah *worm* lokal yang membawa-bawa nama pembawa acara Jejak Petualang di TV7 yaitu Riyanni Djangkaru. *Worm* ini memiliki ukuran *file* sebesar 40 KB dengan *icon* yang disamarkan sehingga terlihat seperti *file* jpg.

Saat *file infector* dieksekusi, *worm* membuat dua *file*, yaitu *file* 'xpshare.exe' pada direktori 'C:\!submit' dan *file* 'riyani_jangkaru.exe' pada root direktori 'C:\', kemudian menambahkan *value* 'winloader' pada *registry* Run yang akan mengaktifkan *worm* setiap Windows *startup*. Hanya saja *worm* ini baru banyak dikenal pada pertengahan Juli 2005.

2.8.4 Kangen

Pada pertengahan April 2005, suatu *worm* lokal kembali meresahkan pengguna komputer, dengan berbekal *refrain* lagu Kangen (Dewa 19) *worm* ini sukses menginfeksi ratusan komputer di Indonesia. Seperti *worm* lokal terdahulu,

Kangen juga dibuat dengan Visual Basic, menggunakan *icon* Microsoft Word dan melakukan penyebaran lewat disket, anehnya *worm* satu ini sama sekali tidak dikompresi.

Saat *file infector* dieksekusi *worm* ini meng-*copy* dirinya ke direktori sistem Windows dengan nama 'CCAPPS.EXE' dengan ukuran *file* sebesar 64 KB, yang menarik dari *worm* ini adalah *worm* akan berusaha menonaktifkan program Task Manager, MS Config dan Registry Editor. Saat buku ini ditulis *worm* Kangen sudah mencapai varian M.

2.8.5 Kumis

Diawal bulan Juli 2005, seorang *worm writer* yang konon terinpirasi oleh seorang dosennya yang berkumis tebal membuat *worm* yang kemudian disebut dengan *worm* Kumis. *Worm* ini berukuran sebesar 76 KB dan dikhususkan untuk sistem operasi Windows XP dan Windows Server 2003.

Saat *file infector* dieksekusi *worm* ini membuat salinan dirinya ke direktori sistem Windows dengan nama 'username logon.exe' dimana *username* adalah nama *user* aktif, kemudian memanipulasi *registry* agar *worm* tetap aktif. Menariknya, *worm* kumis ini akan me-*restart* kompuer setiap kali Windows *startup*.

2.8.6 Decoil

Worm Decoil yang juga sering disebut Decoy ini muncul di penghujung tahun 2005, *worm* lokal ini menyembunyikan *file* dokumen Microsoft Word dan membuat salinan dengan nama yang persis dengan nama *file* yang disembunyikan.

Saat *file infector* dieksekusi, *worm* ini meng-*copy* dirinya ke direktori Windows dengan nama 'Iexplorer.exe', pada direktori sistem Windows dan sub direktori '\i75-d2' dengan nama *file* 'dkernel.exe'. Decoy tergolong *worm* dengan ukuran yang sangat besar yaitu sebesar 154 KB (besar file ini detelah dikompres dengan UPX).

2.8.7 Rontokbro

Diawal bulan Oktober 2005, pengguna komputer di Indonesia kembali dikejutkan oleh sebuah *worm* lokal yang sudah menggunakan smtp sendiri dalam mengirimkan *file infector*-nya dan selektif dalam pemilihan alamat *e-mail target*.

Worm yang diisukan sebagai *worm* lokal ter-anyar yang pernah ada ini ternyata dibuatu dalam bahasa Visual Basic, RontokBro juga memblokir aplikasi Registry Editor dan memonitor *caption* pada aplikasi *browser*, jika suatu string tertentu ditemukan oleh RontokBro maka komputer akan di-*restart* secara otomatis.

2.8.8 Nobron

Tidak begitu lama setelah kemunculan RontokBro, *worm* yang kemudian dikenalai dengan nama W32/Nobron.A@mm oleh Norman Virus Control ini berusaha untuk membersihkan sistem dari *worm* RontokBro, Nobron memiliki ukuran *file* sebesar 84 KB. *Worm* juga berusaha untuk menonaktifkan aplikasi Registry Editor, MS Config, Tas Manager dan CMD.

2.8.9 Runitis

Pada akhir bulan Nopember 2005, kembali diketahui sebuah *worm* lokal dengan ukuran *file* sebesar 164 KB dan telah dikompres dengan aplikasi ASPack.

Worm ini menggunakan nama yang diambil dari nama seorang penyanyi dari Malaysia yaitu Siti Nurhaliza (Runitis jika dibaca terbalik menjadi Sitinur).

Runitis menggunakan *icon* Internet Explorer dalam penyebarannya. *Worm* ini menghapus *file* program regedit.exe (Registry Editor), msconfig.exe (MS Config), wmplayer.exe (Windows Media Player), winamp.exe dan wnampa.exe (Winamp) serta beberapa *file* program lainnya kemudian membuat salinan dirinya dengan nama dan letak yang sama, sehingga *user* malah akan menjalankan *worm* apabila mengeksekusi aplikasi-aplikasi tersebut.

2.8.10 Bluefantasy

Di akhir bulan Januari 2006, sebuah *worm* lokal bernama BlueFantasy ikut berlomba dengan *worm* lainnya, walaupun *worm* yang berukuran 64 KB ini tidak melewati proses enkripsi dan kompresi namun BlueFantasy mempertahankan ciri khas *worm* komputer yang membuatu banyak salinan, dengan cara memonitor direktori aktif dan membuat salinan pada direktori tersebut.

Saat *file infector* dieksekusi BlueFantasy meng-*copy* dirinya ke direktori sistem Windows dengan nama file 'Win32.com' dan menggunakan atribut *Hidden*, selain itu *worm* juga membuat salinan pada direktori Desktop, My Documents dan StartUp. Kemudian *worm* memanipulasi suatu nilai *registry* yang akan membuat *file worm* tetap tereksekusi saat Windows *startup*, dan membuat tipe *file executable* seperti exe dan scr akan terlihat sebagai aplikasi Microsoft Word.

2.9 Diagram Arus Data (*Data Flow Diagram*)

DFD atau singkatan dari *Data Flow Diagram* merupakan representasi grafik dari suatu sistem yang menunjukan proses atau fungsi, aliran data, tempat penyimpan data dan *entitas eksternal*. DFD juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. Dengan menggunakan DFD, rancangan yang akan kita buat akan lebih terarah dan lebih rinci. Sehingga kita tidak akan mengalami kesulitan dalam melakukan perancangan. *Data flow diagram* memiliki empat komponen, antara lain akan dijelaskan dibawah ini. [6]

1. *Terminator (external exitity)*

   Terminator mewakili *entity external* yang berkomunikasi dengan sistem yang sedang dikembangkan. *Terminator* merupakan kesatuan dilingkungan sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*) sumber atau tujuan (*source* dan *sink*). Terminator dapat juga berupa departemen, divisi atau sistem diluar sistem yang berkomunkasi dengan sistem yang dikembangkan.

2. Proses

   Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output.* Setiap proses harus diberikan penjelasan yang lengkap sebagai berikut :

a. Identifikasi Proses

umunya berupa angka yang menunjukan nomor dari proses atau ditulis pada bagian atas simbol proses.

b. Nama Proses

Menunjukan apa yang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya nama proses diletakan dibawah identifikasi proses.

3. Penyimpanan data *(data store)*

Data store digunakan sebagai sarana untuk mengumpulkan data. Data store disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada salah satu ujungnya atau dua garis horizontal saja. *Data store* ini biasanya berkaitan dengan penyimpanan-pemyimpanan seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, contohnya *file* pita magnetic, *file* disket atau *file harddisk*. *Data store* juga berkaitan dengan pemyimpanan data.

4. Alur Data *(Data Flow)*

Suatu data *flow* atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau satu paket data atau informasi dari suatu bagian sistem ke bagian lainnya. Selain menunjukan arah, alur data pada model yang dibuat dapat merepresentasikan bit, karakter, pesan, formulir, bilangan *real* dan macam-macam informasi yang berkaitan dengan komputer.

Ada tiga tipe DFD, yaitu *context diagram* (diagram konteks), *data flow diagram fisik*, dan *data flow diagram logika*. [6]

2.10 Diagram Konteks

Diagram Konteks adalah bagian dari DFD yang berfungsi memetakan model model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Diagram ini adalah diagram *level* tertinggi dari DFD. Diagram konteks menyoroti sejumlah karateristik penting sistem, yaitu :

1. Kelompok Pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
2. Data Masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data Keluar, yaitu data yang dihasilkan sistem dan diberikan kedunia luar.
4. Penyimpanan Data (*storage*), yaitu digunakan secara bersama antara sistem dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunkan oleh sistem. Hal ini berarti pembuatn simbol *data storage* dalam diagram konteks dibenarkan, dengan syarat simbol tersebut merupakan bagian dari dunia diluar sistem.
5. Batasan, antara sistem dan lingkungan.

2.11 *Data Flow Diagram* Fisik

Data flow diagram fisik adalah representasi dari sebuah sistem yang menunjukan *entitas-entitas internal* dan *eksternal* dari sistem tersebut, *Entitas-entitas internal* adalah *personal,* tempat atau mesin dalam sistem tersebut yang

mentransformasi data. Maka data *flow* diagram fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukan dimana, bagaimana, dan siapa proses-proses dalam sebuah sistem dilakukan.

2.12 *Data Flow Diagram Logika*

Data flow diagram logika digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). *Data flow diagram logika* tidak menekankan pada bagaimana sistem diterapkan, tetapi penekanannya hanya pada logika pada kebutuhan-kebutuhan sistem, yaitu proses-proses apa secara logika yang dibutuhakan oleh sistem.

2.13 Jaringan Semantik

Model jaringan sematik merupakan grafik, yang terdiri dari simpul-simpul yang merepresentasikan objek fisik atau objek konsep, dan busur-busur yang menunjukan relasi antara simpul-simpul tersebut. Jaringan semantik merupakan alat efektif untuk merepresentasikan pemetaan data, yang bertujuan mencegah terjadinya duplikasi data. [2]

2.14 MS. Visual Basic 6.0

BASIC, adalah singkatan dari *Beginners' All-purpose Symbolic Instruction Code* adalah sebuah kelompok bahasa pemrograman tingkat tinggi. Secara harfiah, BASIC memiliki arti "kode instruksi simbolis semua tujuan yang dapat digunakan oleh para pemula". Memang, istilah "Bahasa BASIC" di sini juga bisa diartikan menjadi bahasa untuk pemula, atau dengan kata lain, disebut sebagai bahasa dasar, tapi hal tersebut dirasa kurang tepat, mengingat BASIC dapat juga digunakan oleh para pemrogram ahli.

BASIC pertama kali dikembangkan pada tahun 1963 oleh John George Kemeny dan Thomas Eugene Kurtz yang berasal dari Dartmouth Collage, untuk mengizinkan akses terhadap komputer bagi para mahasiswa jurusan selain jurusan ilmu eksakta. Pada waktu itu, hampir semua komputer membutuhkan perangkat lunak, dan waktu itu belum ada perangkat lunak yang dijual secara bebas, sehingga hanya orang-orang tertentulah yang dapat menggunakan komputer, yakni para matematikawan dan ilmuwan, karena mereka dapat membangun perangkat lunak sendiri. Bahasa BASIC, setelah diciptakan menjadi menjamur dan banyak dimodifikasi. Bahasa BASIC menjadi bahasa yang paling populer digunakan pada komputer mikro pada akhir tahun 1970-an dan komputer rumahan pada tahun 1980-an. Dan hingga saat ini, menjadi bahasa yang dialeknya beberapa kali berevolusi.

Bill Gates, pendiri Microsoft, memulai bisnis softwarenya dengan mengembangkan interpreter bahasa Basic untuk Altair 8800, untuk kemudian ia ubah agar dapat berjalan di atas IBM PC dengan sistem operasi DOS. Perkembangan berikutnya ialah diluncurkannya BASICA (basic-advanced) untuk DOS. Setelah BASICA, Microsoft meluncurkan Microsoft QuickBasic dan Microsoft Basic (dikenal juga sebagai Basic Compiler).

Sejarah BASIC di tangan Microsoft sebagai bahasa yang diinterpretasi (BASICA) dan juga bahasa yang dikompilasi (BASCOM) membuat Visual Basic diimplementasikan sebagai gabungan keduanya.

Programmer yang menggunakan Visual Basic bisa memilih kode terkompilasi atau kode yang harus diinterpretasi sebagai hasil executable dari

kode VB. Sayangnya, meskipun sudah terkompilasi jadi bahasa mesin, DLL bernama MSVBVMxx.DLL tetap dibutuhkan. Namun karakteristik bahasa terkompilasi tetap muncul (ia lebih cepat dari kalau kita pakai mode terinterpretasi).

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang bersifat event driven dan menawarkan Integrated Development Environment (IDE) visual untuk membuat program aplikasi berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman Common Object Model (COM). Visual Basic merupakan turunan bahasa BASIC dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan Data Access Objects (DAO), Remote Data Objects (RDO), atau ActiveX Data Object (ADO), serta menawarkan pembuatan kontrol ActiveX dan objek ActiveX. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi eksternal tambahan.

Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62%

pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript, C#, dan Java.

Visual Basic merupakan bahasa yang mendukung OOP, namun tidak sepenuhnya. Beberapa karakteristik objek tidak dapat dilakukan pada Visual Basic, seperti *Inheritance* tidak dapat dilakukan pada class module. *Polymorphism* secara terbatas bisa dilakukan dengan mendeklarasikan *class module* yang memiliki *Interface* tertentu. Visual Basic (VB) tidak bersifat case sensitif.