BAB II
TEORI PENUNJANG

2.1 Sistem Distribusi

Pendistribusian bantuan merupakan mekanisme atau prosedur yang menghubungkan antara pemberi bantuan (donator) dan penerima bantuan (korban bencana) pada suatu situasi kebencanaan. Bantuan dalam hal ini adalah bantuan kemanusiaan yang terdiri dari tempat penampungan sementara, bantuan pangan, sandang, air bersih dan sanitasi, serta pelayanan kesehatan. Pengorganisasian atau tata cara pemberian bantuan pemenuhan kebutuhan dasar dikoordinasikan oleh Badan Nasional Penanggulangan Bencana (BNPB) serta Badan Penanggulangan Bencana Daerah (BPBD) sesuai dengan tingkatan bencana, melalui Sistem Komando Tanggap Darurat Bencana[1].

Pendistribusian bantuan yang telah berlangsung selama ini memiliki beberapa kebijakan. Kebijakan tersebut yaitu pertama, penanggulangan bencana dilaksanakan secara terpadu dan terkoordinasi yang melibatkan seluruh potensi pemerintah, swasta dan masyarakat, baik pada tahap pra bencana, saat terjadi bencana maupun pasca bencana. Kedua, memberikan penjaminan pemenuhan hak masyarakat korban bencana dan pengungsi yang terkena bencana terutama pelayanan kebutuhan dasar secara adil.

Dalam pelaksanaan pendistribusian atau pemberian bantuan terdapat beberapa langkah, diantaranya:

1. Penyusunan Daftar Penerima Bantuan

   Agar bantuan yang diberikan tepat sasaran, maka perlu dilakukan identifikasi penerima bantuan secara rinci sehingga tidak satu orangpun korban bencana yang tertinggal dan tidak menerima bantuan. Langkah-langkah yang dilakukan pada kegiatan pendaftaran penerima bantuan adalah sebagai berikut:

   a) Mengidentifikasi jumlah keluarga di setiap titik penampungan.
   b) Berdasarkan identifikasi jumlah keluarga, disusun daftar penerima bantuan secara keseluruhan sesuai dengan kelompok umur.

2. Penilaian Kebutuhan

   Agar bantuan yang diberikan pada korban sesuai dengan yang dibutuhkan, perlu dilakukan penilaian dengan mempertimbangkan kelompok umur, jenis kelamin, dan kelompok rentan lainnya. Susunan daftar penerima bantuan dapat dijadikan data awal dalam mempertimbangkan kebutuhan penerima bantuan. Langkah-langkah yang dilakukan pada kegiatan penilaian kebutuhan adalah sebagai berikut:

   a) Mengidentifikasi kebutuhan penerima bantuan (korban bencana).
   b) Mengidentifikasi sumber barang-barang/aset yang masih dimiliki korban/penerima bantuan.
   c) Menentukan jenis bantuan yang diperlukan penerima bantuan (korban bencana).

3. Penilaian Jenis Bantuan

   Langkah berikutnya menentukan jumlah bantuan yang harus didistribusikan pada seluruh penerima bantuan (korban bencana). langkah-langkahnya sebagai berikut:

   a) Menyiapkan daftar penerima bantuan dan daftar kebutuhan yang diperlukan sesuai dengan prioritas.
   b) Menghitung perkiraan jumlah yang harus didistribusikan pada setiap hari/minggu/bulan.
   c) Menghitung jumlah bantuan tidak terduga.
   d) Menghitung jumlah keseluruhan bantuan yang diperlukan selama masa di penampungan.

4. Pendistribusian Bantuan

   Pendistribusian bantuan harus cepat dan tepat serta sesuai dengan kondisi setempat, dengan melalui beberapa mekanisme:

   a) Penerima bantuan pangan diidentifikasi dan menjadi sasaran berdasarkan kebutuhan.
   b) Metode distribusi dirancang melalui konsultasi dengan kelompok setempat, lembaga-lembaga mitra, dan melibatkan berbagai kelompok penerima.

c) Titik-titik distribusi sedekat mungkin dengan hunian sementara penerima untuk memastikan akses yang mudah dan aman.
d) Kualitas, jumlah jatah makanan/pangan dan rencana distribusi diinformasikan jauh sebelumnya kepada penerima bantuan.
e) Kinerja dan efektifitas program bantuan pangan dimonitor dan dievaluasi dengan semestinya.

5. Pencatatan dan Pelaporan

   Kegiatan pencatatan dan pelaporan yang berkaitan dengan mekanisme pemberian bantuan mulai dari setiap tahap didokumentasikan ataupun dicatat dalam suatu dokumen sebagai bukti pertanggungjawaban sebagai berikut:

   a) Pencatatan penerimaan bantuan meliputi: pemberi bantuan, jumlah, dan jenis bantuan, serta waktu penyerahan bantuan.
   b) Pencatatan penyaluran meliputi: penerima bantuan, jumlah, dan jenis bantuan, waktu penyaluran, lokasi penyaluran bantuan, serta penanggungjawab (contact persons).
   c) Pencatatan persediaan logistik dan peralatan.
   d) Pelaporan hasil penerimaan dan penyaluran bantuan disampaikan kepada SATLAK PB/BPBD kabupaten/kota, SATKORLAK PB/BPBD provinsi atau BNPB dengan tembusan lembaga/instansi yang memberi bantuan.

2.2 Bencana

Bencana adalah peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam dan/atau faktor nonalam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda, dan dampak psikologis.

Jenis-jenis bencana antara lain sebagai berikut[2]:

1. Bencana alam adalah suatu peristiwa alam yang mengakibatkan dampak besar bagi populasi manusia. Peristiwa alam dapat berupa banjir, letusan gunung berapi, gempa bumi, tsunami, tanah longsor, badai salju, kekeringan, hujan es, gelombang panas, badai tropis, tornado, kebakaran

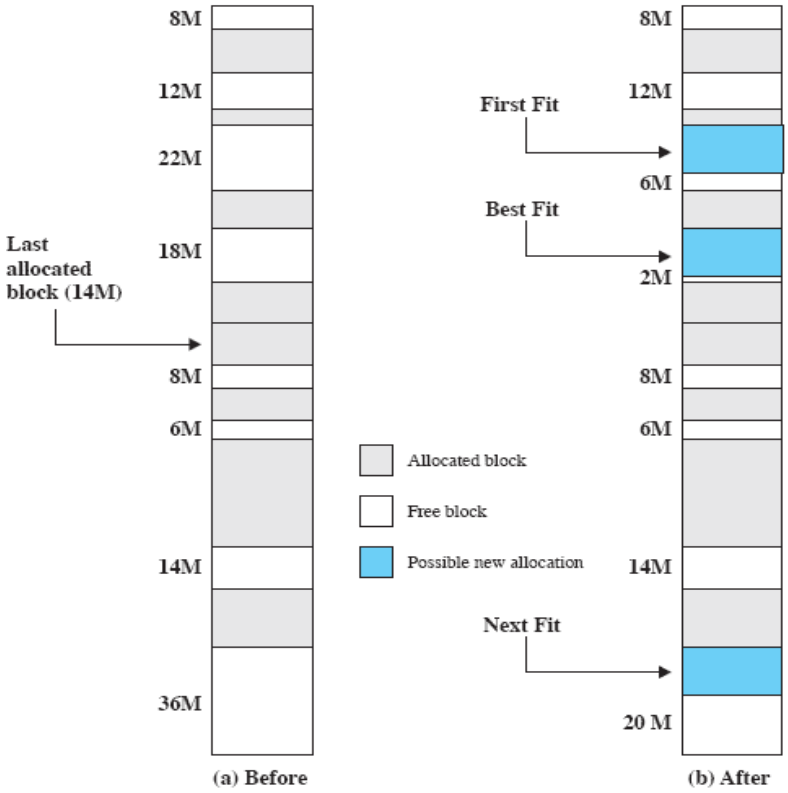
liar dan wabah penyakit. Bencana alam dibagi menjadi dua jenis berdasarkan penyebabnya yaitu bencana geologis dan klimatologis.

a) Bencana alam geologis adalah bencana yang disebabkan oleh gaya-gaya yang berasal dari dalam bumi (gaya endogen). Yang termasuk dalam bencana alam geologis adalah gempa bumi, letusan gunung berapi, dan tsunami.
b) Bencana alam klimatologis merupakan bencana alam yang disebabkan oleh faktor angin dan hujan. Contoh bencana alam klimatologis adalah banjir, badai, banjir bandang, angin puting beliung, kekeringan, dan kebakaran alami hutan (bukan oleh manusia). Gerakan tanah (longsor) termasuk juga bencana alam, walaupun pemicu utamanya adalah faktor klimatologis (hujan), tetapi gejala awalnya dimulai dari kondisi geologis (jenis dan karakteristik tanah serta batuan dan sebagainya).

2. Bencana nonalam adalah bencana yang diakibatkan oleh peristiwa atau rangkaian peristiwa nonalam yang antara lain berupa gagal teknologi, gagal modernisasi, epidemi, dan wabah penyakit.
3. Bencana sosial adalah bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang diakibatkan oleh manusia yang meliputi konflik sosial antar kelompok atau antar komunitas masyarakat, dan teror.

2.3 Algoritma First-fit

Algoritma *First-fit* yaitu manajer memori mencari (men-*scan*) sepanjang list segmen sampai mendapatkan rongga atau tempat yang cukup besar[3]. Rongga tersebut dibagi ke dalam dua bagian, satu untuk proses dan satu lagi untuk ruang kosong (*unused memory*), kecuali jika proses memerlukan seluruh rongga maka rongga tersebut hanya dipakai untuk proses. Dalam sistem distribusi ini, algoritma *first fit* akan menentukan kemana bantuan dari donator disalurkan kepada posko pengungsian yang membutuhkan sesuai dengan besarnya jumlah materi bantuan, jumlah korban di setiap posko, dan jumlah kebutuhan materi sehari-hari.



Gambar 2.1 Contoh Manajemen Memori Sebelum dan Sesudah Alokasi 16 MByte Blok

Gambar 2.1 menunjukan algoritma first fit akan men-scan memori dari awal, kemudian memilih blok dengan space 22 MByte, menyisakan 6 MByte untuk ruang kosong[4].

2.4 Website

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi[5].

Secara terminologi, website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP (Hypertext Transfer Protocol), yaitu

protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari website bisa diakses melalui sebuah URL (*Unicode Resource Locator*) yang biasa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka sususan keseluruhan dan bagaimana arus informasi ini berjalan.

Penemu website adalah Sir Timothy John Tim Berners-Lee, sedangkan website yang tersambung dengan jaringan, pertamakali muncul pada tahun 1991. Maksud dari Tim ketika membuat website adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan secara gratis oleh semua orang.

Sebuah website bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya website itu menujukkan beberapa topik khusus, atau kepentingan tertentu. Sebuah website bisa berisi *hyperlink* (pranala) yang menghubungkan ke website lain, jadi, kadangkala perbedaan antara website yang dibuat oleh individu perseorangan dengan website yang dibuat oleh organisasi bisnis bisa saja tidak kentara.

Website ditulis atau secara dinamik di konversi menjadi HTML dan diakses melalui sebuah program software yang biasa disebut dengan *web browser*, yang dikenal juga dengan HTTP *Client*. Halaman web dapat dilihat atau diakses melalui jaringan komputer dan internet, perangkatnya bisa saja berupa komputer pribadi, laptop, ataupun telepon selular.

2.5 UML

UML (Unified Modeling Language) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak[6].

2.5.1 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian di saat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shaler dan Stephen Mellor (Shaler-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen, 1991), dan masih banyak lagi.

Buku terkenal yang juga berkembang selanjutnya adalah karangan Ivar Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya. Didalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses mengenai metodologi berorientasi objek.

Karena banyaknya metodologi-metodologi yang berkembang pesat itu, maka muncul ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar

Jacobson, dan beberapa konsep lainya dimana James R. Rumbaigh, Grandy Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML).

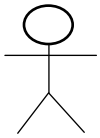
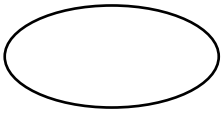
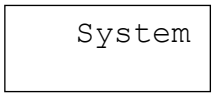
2.5.2 Diagram UML

Berikut akan dijelaskan diagram yang akan digunakan dalam pembangunan sistem monitoring distribusi bantuan berbasis website ini, yaitu *use case diagram*, *class diagram*, *sequence diagram*, dan a*ctivity diagram*.

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk (*behavior)* aplikasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak mengguankan fungsi-fungsi itu. Penjelasan tentang simbo-simbol *use case* dapat dilihat pada tabel 2.1.

Tabel 2.1 Definisi Simbol Use Case Diagram

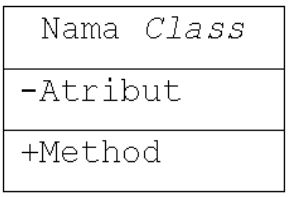
Simbol	Nama Simbol	Fungsi
	Aktor	Merupakan orang, proses, atau sistem lain yang berinteraksi dengan aplikasi yang akan dibuat di luar aplikasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang
	*Use Case*	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
	Association	Merelasikan Aktor dengan *use case*
	*System Boundary*	Menggambarkan batasan sistem terhadap lingkungannya

2. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut, merupakan variable-variabel yang dimiliki oleh satu kelas.
2. Operasi, atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Class diagram dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sesuai. Contoh *class diagram* dapat dilihat pada gambar 2.2.

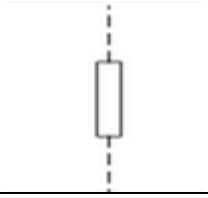
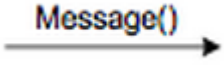


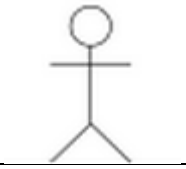
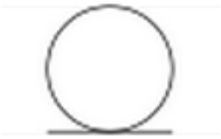
Gambar 2.2 Contoh Class Diagram

3. Sequence Diagram

Sequence diagram menggambarkan *behavior* objek pada *use case* dengan mendeskripsikan waktu hidup dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan melihat scenario yang ada pada use case. Penjelasan tentang simbol-simbol *sequence diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Definisi Simbol Sequence Diagram

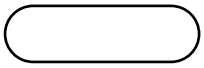
Simbol	Nama Simbol	Fungsi
	*Lifelines*	Menggambarkan waktu mulai dan berakhirnya sebuah message
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktfitas yang terjadi

	*Actor*	Digunakan untuk menggambarkan user / pengguna
	*Boundary*	Digunakan untuk menggambarkan sebuah form atau antarmuka
	*Control Class*	Digunakan untuk menghubungkan boundary dengan entity
	*Entity Class*	Menyimpan data/informasi yang digunakan dalam perangkat lunak

4. Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Tabel 2.3 adalah penjelasan tentang symbol-simbol *activity diagram.*

Tabel 2.3 Definisi Simbol Activity Diagram

Simbol	Nama Simbol	Fungsi
	*Initial State*	Pernyataan awal dari sebuah aktivitas
	*Final State*	Pernyataan akhir dari sebuah aktivitas
	Transisi	Menggambarkan alur dari aksi – aksi yang ada pada aktivitas
	Aksi	Menggambarkan aksi yang ada pada aktivitas

2.6 PHP

PHP (*Hypertext Preprocessor)* yaitu bahasa pemrograman web server-side yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML

dan berada pada server (*server side* HTML *embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua script PHP dieksekusi pada server dimana *script* tersebut dijalankan.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP (*Hypertext Preprocessing)*.

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek[7].

2.7 Barcode

Sebuah kode batang atau kode palang (bahasa Inggris: *barcode*) adalah suatu kumpulan data optik yang dibaca mesin. Kode batang ini mengumpulkan data dalam lebar (garis) dan spasi garis paralel dan dapat disebut sebagai kode batang atau simbologi linear atau 1D (1 dimensi). *Barcode* menyimpan data-data spesifik seperti kode produksi, nomor identitas, dan lain-lain sehingga sistem komputer dapat mengidentifikasi dengan mudah informasi yang dikodekan dalam *barcode*[8].

Komputer tidak secara langsung dapat membaca data yang terkandung dalam kode bar tersebut, oleh karena itu sebelumnya kode yang ada harus ditangkap dan diterjemahkan ke dalam format data yang dapat dibaca oleh komputer. Alat yang dapat membaca dan mengirimkannya ke dalam komputer itulah yang disebut *Barcode Reader* atau yang biasa disebut *Barcode Scanner*.

Seperangkat *Barcode Scanner* terdiri dari *scanner*, *decoder* dan kabel yang menyambungkan *decoder* dengan komputer. *Barcode Scanner* tersebut memindai simbol, menangkap dan merubah kode bar menjadi data elektrik lalu mengirimkannya ke komputer dengan format data yang sederhana[9].