

Implementasi Algoritma Pohon Triner untuk Perhitungan Bonus Jaringan MLM Berbasis Web

Yendy Winata

Program Studi Teknik Informatika, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudiro Husodo no 5-25, Yogyakarta 55224, Indonesia
Email: yendyw@gmail.com

Abstract. *Implementation of Triner Tree Algorithm for Calculation of Web-Based MLM Network Bonus.* In the development of the marketing world these days there is a new term that is Multi Level Marketing (MLM). In running an MLM, there is a calculation of bonuses that have a specific formula to attract consumers to become members as well as to increase the income of the MLM itself. To perform a calculation of this bonus, if done manually would be very time consuming, because the administrator will first look at each member and the tree structure that, then he can do the bonus calculation network. On the other hand, the managers of MLM are required to present the information these bonuses quickly and accurately to its members. Would be made an auxiliary program to calculate it with a web-based so that members can get information about bonuses own network quickly. This system uses a tree structure that is triner tree structure having subtrees maximum of 3 pieces. Web-based system is expected to convey information to members with the bonus quickly.

Keywords: *triner tree algorithm, the bonus network, MLM*

Abstrak. Dalam perkembangan dunia pemasaran akhir-akhir ini terdapat suatu istilah baru yaitu Multi Level Marketing (MLM). Dalam menjalankan suatu MLM ini, terdapat suatu perhitungan bonus yang mempunyai rumus tertentu untuk menarik minat para konsumen agar menjadi anggotanya sekaligus untuk meningkatkan pendapatan dari MLM itu sendiri. Untuk melakukan suatu perhitungan bonus ini, jika dilakukan secara manual akan sangat memakan waktu, karena administrator akan melihat dulu setiap anggota dan struktur pohonnya itu, barulah dia bisa melakukan perhitungan bonus jaringan. Disisi lain, pengelola MLM tersebut dituntut untuk menyajikan informasi bonus-bonus tersebut secara cepat dan akurat kepada para anggotanya. Untuk itu akan dibuat suatu program bantu untuk menghitungnya dengan berbasis *web* agar para anggota dapat memperoleh informasi tentang bonus jaringannya sendiri secara cepat. Sistem ini menggunakan struktur pohon triner yaitu struktur pohon yang mempunyai sub pohon maksimal 3 buah. Sistem berbasis web ini diharapkan dapat menyampaikan informasi bonus kepada anggota dengan semakin cepat.

Kata Kunci: algoritma pohon triner, bonus jaringan, MLM

1. Pendahuluan

1.1. Latar Belakang Masalah

Dalam perkembangan dunia pemasaran akhir-akhir ini terdapat suatu istilah baru yaitu Multi Level Marketing (MLM). Dalam menjalankan MLM ini, terdapat suatu perhitungan bonus yang mempunyai rumus tertentu untuk menarik minat para konsumen agar menjadi anggotanya sekaligus untuk meningkatkan pendapatan dari MLM itu sendiri.

Dalam penelitian ini, MLM yang ditangani adalah MLM yang bergerak dalam bidang

penjualan *voucher* isi ulang untuk *handphone* GSM. MLM ini menerapkan suatu perhitungan bonus yang bernama bonus jaringan. Bonus jaringan ini hanya bisa didapat jika orang tersebut menjadi anggota. Selain itu, syarat untuk mendapatkan bonus jaringan ini, dia juga harus mempunyai bawahan. Jika orang tersebut membeli *voucher* dan bawahannya membeli *voucher*, barulah dia mendapatkan bonus jaringan tersebut. Perhitungan bonus jaringan dalam MLM yang ditangani ini dilakukan secara berkala, yaitu setiap akhir bulan.

Di dalam MLM yang ditangani ini terdapat suatu struktur, yang mana jika struktur ini digambarkan, akan membentuk hubungan antara atasan dan bawahannya. Struktur ini bentuknya sama seperti struktur pohon pada struktur data. Struktur pohon ini tidak menggambarkan hubungan pada tingkatan yang sama. Untuk menghitung bonus jaringan tersebut, struktur pohon ini sangat dibutuhkan. Untuk melakukan suatu perhitungan bonus ini, jika dilakukan secara manual akan sangat memakan waktu, karena administrator akan melihat dulu setiap anggota dan struktur pohonnya itu, barulah dia bisa melakukan perhitungan bonus jaringan. Disisi lain, pengelola MLM tersebut dituntut untuk menyajikan informasi bonus-bonus tersebut secara cepat dan akurat kepada para anggotanya.

1.2. Rumusan Masalah

Melihat latar belakang masalah yang telah diuraikan diatas, penulis dapat menekankan merumuskan permasalahan yang menjadi penekanan dalam penelitian ini, yaitu (1) Bagaimana membuat program bantu yang mampu melakukan perhitungan bonus untuk suatu sistem yang berbentuk pohon dengan berbasis *web*? (2) Bagaimana membuat program yang mampu menampilkan visualisasi struktur yang berbentuk pohon dalam suatu *web*? (3) Bagaimana membuat program yang mampu menampilkan *downline* (bawahannya) dari anggota sampai dengan kedalaman 10 level?

1.3. Batasan Masalah

Untuk penulisan tugas akhir ini, penulis membatasi permasalahannya adalah sebagai berikut: (1) Program ini dibuat hanya untuk melakukan perhitungan bonus jaringan saja, tidak menangani pembelian produk dan pendaftaran anggota. (2) Perhitungan bonus jaringan dan penampilan struktur pohon dibatasi sampai dengan kedalaman level 10. (3) Setiap anggota hanya diperkenankan mempunyai paling banyak 3 *downline* (bawahan) untuk 1 level dibawahnya (berbentuk pohon triner).

2. Landasan Teori

2.1. Pohon

Pohon adalah struktur berisi sekumpulan elemen dimana salah satu elemen adalah akar (*root*) dan elemen-elemen lain adalah bagian-bagian pohon yang membentuk susunan hirarki dengan akar sebagai awal mula.¹ Pohon mempunyai bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan *one to many*) antara elemen-elemen. Pohon bisa didefinisikan sebagai kumpulan simpul/*node* dengan satu elemen khusus yang disebut *root* dan *node* lainnya terbagi menjadi himpunan-himpunan yang saling tak berhubungan satu sama lain (disebut sub-pohon).

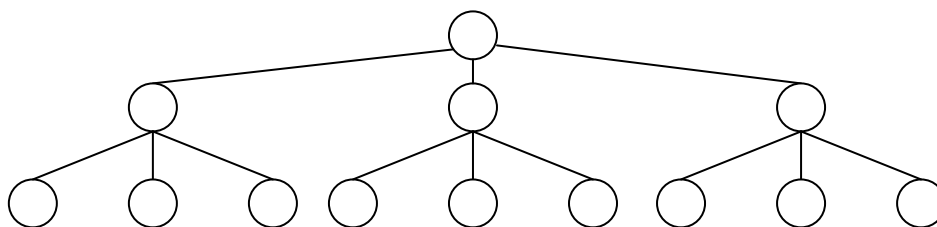
Pohon biner bisa didefinisikan sebagai suatu kumpulan simpul yang mungkin kosong atau mempunyai akar dan paling banyak dua sub-pohon yang saling terpisah yang disebut dengan sub-pohon kiri dan sub-pohon kanan. Penelusuran dalam pohon biner ada 4 macam yaitu: (1) *Inorder*, dengan langkah: Telusuri sub-pohon kiri, Kunjungi node akar, dan Telusuri sub-pohon kanan, (2) *Preorder*, dengan langkah: Kunjungi node akar, Telusuri sub-pohon kiri,

Telusuri sub-pohon kanan, (3) *Postorder*, dengan langkah: Telusuri sub-pohon kiri, Telusuri sub-pohon kanan, Kunjungi node akar, (4) Tingkat demi tingkat dimana penelusuran dimulai dari root, kemudian semua simpul satu level lebih bawah dari root, kemudian semua simpul dua level lebih bawah, dan seterusnya.

2.1.1. Pohon *K*-ary

Pohon *K*-ary adalah pohon yang memiliki *K* subpohon (Shaffer, 1997). Jadi, pohon biner adalah pohon *2*-ary. Pohon *K*-ary memiliki jumlah *node* yang tetap sebanyak *K* buah. Banyak sifat dari pohon biner yang juga merupakan sifat dari pohon *K*-ary ini. Perbedaan yang paling besar hanyalah pada jumlah *node*.

Untuk pengimplementasian dalam perhitungan bonus jaringan MLM ini, akan digunakan suatu algoritma yang mengadaptasi algoritma pohon *K*-ary ini. Algoritma dari pohon *K*-ary ini yang lebih dikenal dengan bentuk yang sederhana yaitu pohon biner akan dikembangkan sehingga pohon tersebut tidak hanya memiliki 2 subpohon, tetapi memiliki 3 subpohon yaitu subpohon kiri, subpohon tengah, subpohon kanan. Dalam hal ini, disebut dengan pohon triner. Untuk lebih jelasnya, pohon triner tersebut akan ditunjukkan pada gambar 1.



Gambar 1. Contoh Struktur Pohon Triner

2.2. Bonus Jaringan MLM

Untuk dapat menerima bonus jaringan dalam MLM ini, terdapat beberapa syarat yang harus dipenuhi oleh anggota, yaitu : (1) Anggota harus membeli minimal 1 buah *voucher*. (2) Bawahan dari anggota harus ada yang membeli *voucher*. (3) Jika dalam satu bulan anggota tersebut tidak membeli *voucher*, maka bonus jaringan ini akan hangus dan dihitung ulang lagi pada bulan berikutnya. Aturan perhitungan bonus jaringan ini adalah sebagai berikut : (1) Bawahan anggota yang berada di level ganjil, setiap pembelian 1 buah *voucher* akan diberi bonus Rp. 100,-. (2) Bawahan anggota yang berada di level genap, setiap pembelian 1 buah *voucher* akan diberi bonus Rp. 500,-.

Sebagai contoh, akan digunakan data yang ada pada gambar 1. Pada gambar tersebut, anggota yang bernama a mempunyai anggota sebanyak 12 orang. Pada gambar tersebut, bawahan anggota yang ber-level ganjil adalah b, c, d. Sedangkan yang ber-level genap adalah e, f, ..., l, n. Jika dibawah anggota e, f, ..., l, n terdapat anggota lagi, maka anggota tersebut ber-level ganjil.

Untuk contoh perhitungan bonus jaringan, jika b, c, d masing-masing membeli 1 buah *voucher*, maka pada level ganjil terdapat 3 buah *voucher*. Jadi bonus pada level ganjil adalah $3 \times \text{Rp. } 100,- = \text{Rp. } 300,-$. Jika e, f, ..., l, n masing-masing membeli 1 buah *voucher*, maka pada level genap terdapat 9 buah *voucher*. Jadi bonus pada level genap adalah $9 \times \text{Rp. } 500,- = \text{Rp. } 4.500,-$. Total bonus jaringan yang diterima pada bulan tersebut adalah $\text{Rp. } 300,- + \text{Rp. } 4.500,- = \text{Rp. } 4.800,-$. Bonus jaringan ini dapat diterima jika a membeli minimal 1 buah *voucher* pada bulan tersebut. Jika tidak, maka bonus jaringan pada bulan tersebut hangus dan akan dihitung ulang lagi pada bulan berikutnya.

Pada bonus jaringan ini diberlakukan sistem *push-up*, yaitu suatu sistem dimana jika ada anggota yang tidak membeli *voucher*, maka anggota yang berada pada posisi paling kiri

yang membeli *voucher* akan naik menggantikan posisi anggota yang tidak membeli *voucher* hanya dalam perhitungan bonus jaringan ini, sedangkan struktur pohon yang sebenarnya masih tetap seperti gambar semula. Untuk contoh akan dipakai data struktur pohon gambar 2.8. Dengan struktur pohon tersebut, anggota b tidak membeli *voucher*, sedangkan yang lainnya masing-masing membeli 1 buah *voucher*. Untuk perhitungan bonusnya ini, karena b tidak membeli *voucher*, maka e akan naik menggantikan posisi b untuk sementara. Jadi bonus yang diterima oleh a adalah sebagai berikut : pada level ganjil $3 \times \text{Rp. } 100,- = \text{Rp. } 300,-$. Pada level genap $8 \times \text{Rp. } 500,- = \text{Rp. } 4.000,-$. Jadi bonus yang akan diterima oleh a adalah Rp. 4.300,-.

3. Perancangan Sistem

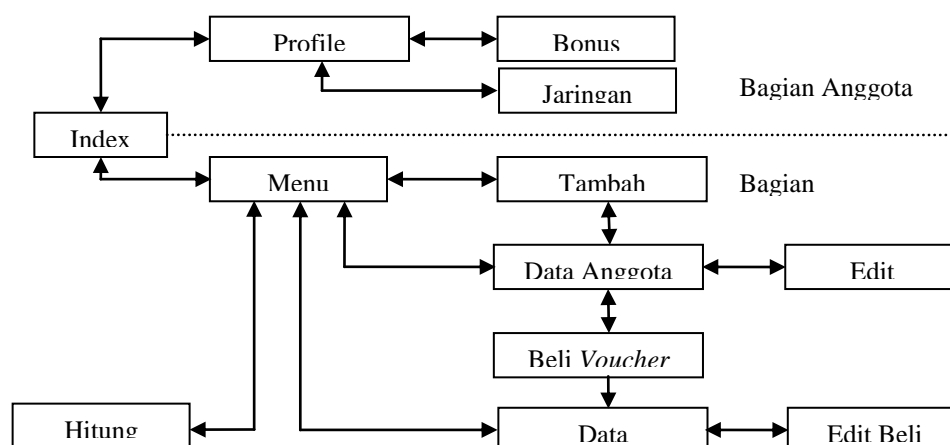
3.1. Perancangan Input

Dalam perancangan *input* ini, terdapat beberapa 2 jenis *input*. Antara lain adalah : *input* untuk pendaftaran anggota baru yang berupa formulir pendaftaran dan *input* untuk pembelian yang berupa nota pembelian. Rancangan *input* anggota baru berfungsi untuk mendapatkan data-data anggota baru. Data-data ini berupa identitas anggota dan juga nomor rekening bank untuk transfer bonus jaringan. Formulir ini diisi oleh anggota baru tersebut yang kemudian akan diserahkan ke perusahaan dan disimpan kedalam database anggota. Untuk rancangan *input* anggota baru data-data yang dibutuhkan antara lain adalah sebagai berikut: (1) No Pohon, berfungsi untuk mengetahui posisi anggota dalam suatu jaringan, sekaligus untuk mengetahui jaringan dari anggota tersebut. (2) No Anggota. (3) Nama Anggota. (4) Alamat Anggota. (5) Tempat Lahir. (6) Tanggal Lahir. (7) Email. (8) No SIM/KTP. (9) No Rekening. (10) Nama Pemilik Rekening. (11) Nama Bank. (12) No HP (*handphone*). (13) *Password*.

Nota pembelian berfungsi untuk mendapatkan data tentang jumlah *voucher* yang dibeli oleh anggota, dimana jumlah *voucher* ini akan dipakai dalam perhitungan bonus jaringan. Nota pembelian ini diisi oleh administrator jika terjadi transaksi pembelian yang dilakukan oleh anggota. Adapun data data yang dibutuhkan pada rancangan *input* untuk pembelian ini adalah: (1) No Nota. (2) Tanggal Pembelian. (3) No Anggota . (4) Jumlah *Voucher*.

3.2. Perancangan Proses

3.2.1. Diagram Struktur Site



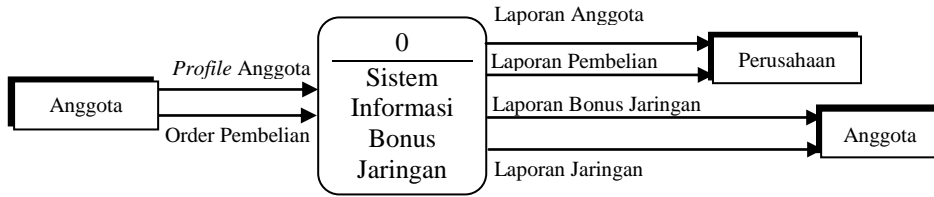
Gambar 2. Diagram Struktur Site

Perancangan struktur dan relasi antar halaman akan menggunakan teknik *mixed hierarchy*. Masing-masing halaman akan memiliki fungsi tersendiri yang saling berhubungan. Halaman index merupakan halaman utama yang akan diakses pertama kali oleh *user*. Jika *user*

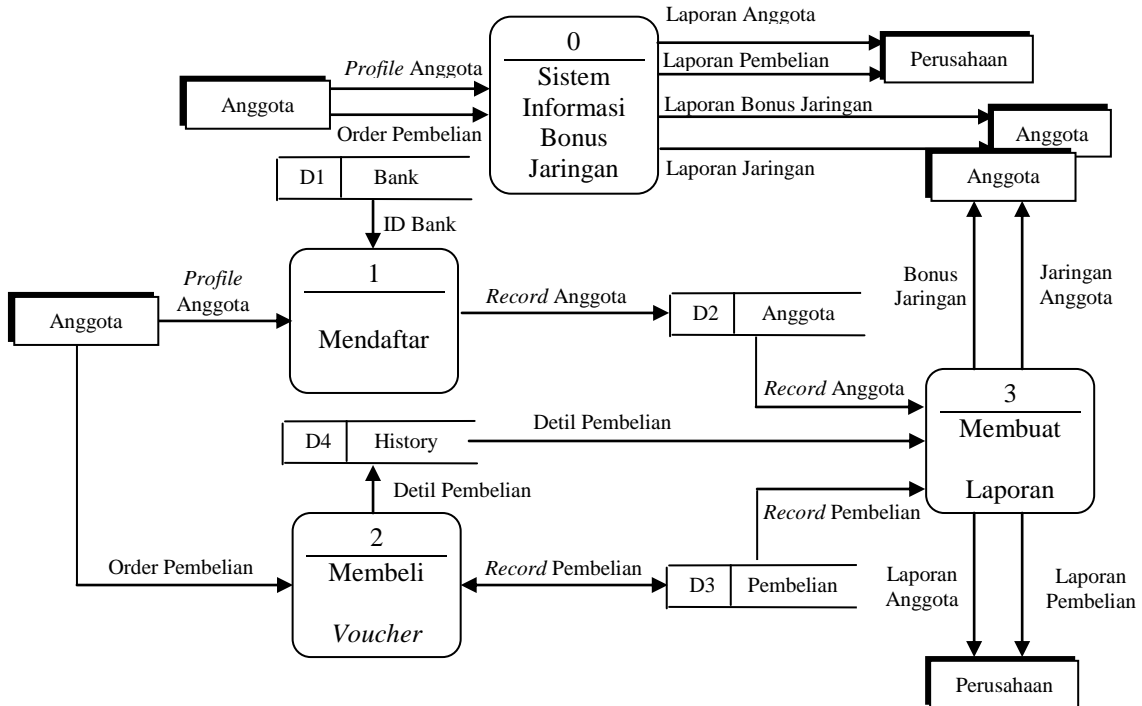
adalah anggota, akan masuk ke bagian anggota, dimana yang tampil adalah *profile* dari anggota tersebut. Jika *user* adalah administrator, maka akan masuk ke bagian administrator, dimana yang tampil adalah menu.

3.2.2. DFD (Data Flow Diagram / Diagram Aliran Data)

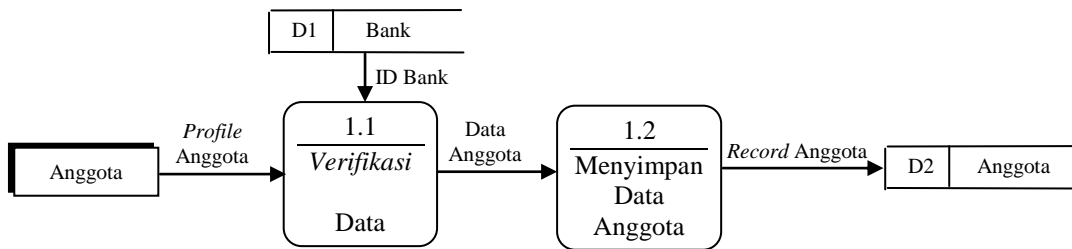
Diagram Aliran Data adalah alat memodelkan proses yang memperhatikan aliran data pada satu sistem dan bekerja dalam sistem tersebut. Diagram Aliran Data menunjukkan aliran data pada sistem.



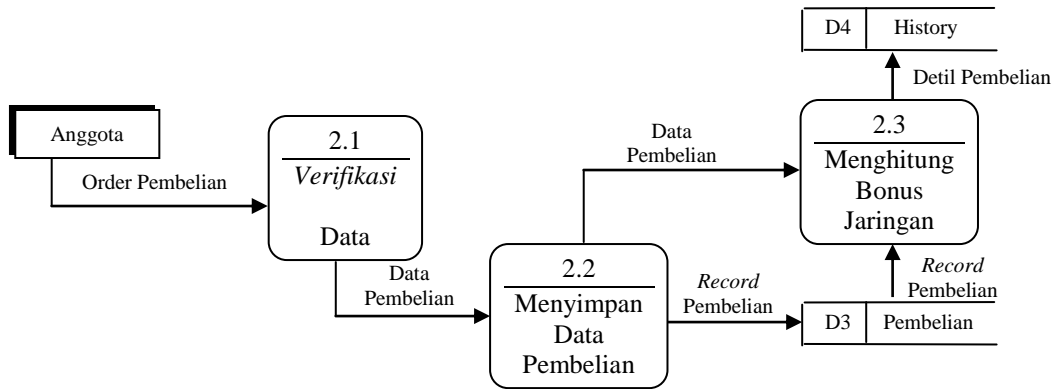
Gambar 3. Diagram Level Konteks



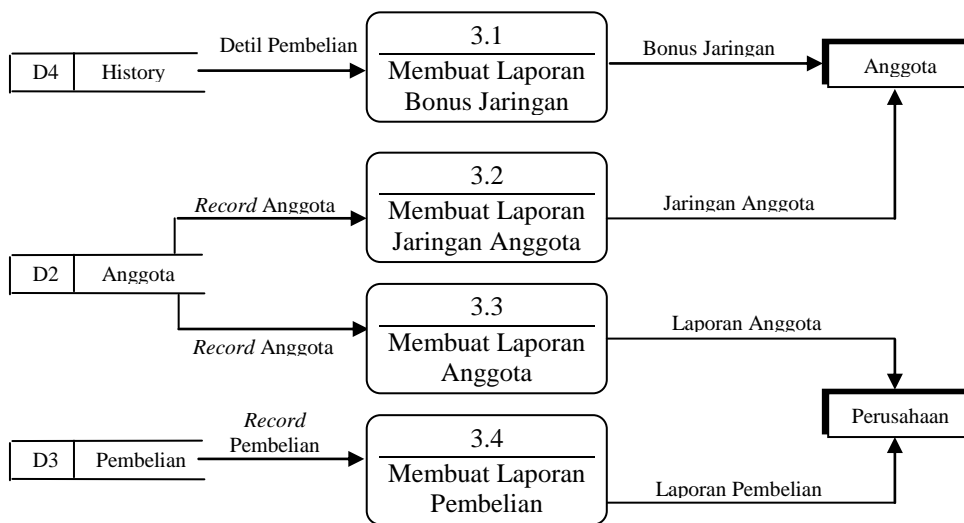
Gambar 4. DAD Level 0



Gambar 5. DAD Level 1 Untuk Mendaftar



Gambar 6. DAD Level 1 Untuk Membeli Voucher



Gambar 7. DAD Level 1 Untuk Membuat Laporan

3.3. Perancangan Halaman Web

3.3.1. Halaman Bonus Jaringan

Halaman bonus jaringan ini akan tampil jika anggota memilih navigasi untuk anggota bagian lihat bonus. Halaman bonus jaringan ini berisi tentang bonus jaringan yang didapat oleh anggota per bulan. Halaman bonus ini hanya menampilkan bonus bulan terakhir yang didapat oleh anggota. Bentuk informasi dari bonus jaringan ini berupa tabel seperti yang terlihat pada gambar 8 yang akan ditampilkan pada bagian konten.

Bonus Bulan xx-xxxx			
No Anggota: xxxxxxxx		Nama:xxxxxxxxxxxxxx	
Level	Jumlah Pembelian	Bobot	Jumlah Bonus
1	xx	Rp. 100,-	Rp. xx,-
2	xx	Rp. 500,-	Rp. xx,-
3	xx	Rp. 100,-	Rp. xx,-
4	xx	Rp. 500,-	Rp. xx,-
.....
Total Bonus			Rp.xx,-

Gambar 8. Halaman Bonus Jaringan

3.3.2. Halaman Hitung Bonus

Halaman ini akan tampil pada bagian konten jika *user* memilih navigasi untuk administrator bagian hitung bonus. Halaman ini digunakan oleh administrator untuk menghitung bonus jaringan. Pada halaman ini terdapat 2 *optionbox* dan 1 tombol. *Optionbox* ini digunakan untuk memasukkan periode perhitungan bonus. Tombol digunakan untuk melakukan proses perhitungan. Setelah tombol ditekan, akan tampil anggota yang mendapatkan bonus. Rancangan halaman ini dapat dilihat pada gambar 9.

Masukkan Periode

Member yang mendapat bonus adalah :

No Anggota	Nama	Jumlah Bonus
xxxxxxxxx	xxxxxxxxxxxxxxxxx	00000
xxxxxxxxx	xxxxxxxxxxxxxxxxx	00000

1 2 3 [Next](#)

Gambar 9. Halaman Hitung Bonus

3.4. Perancangan Kebutuhan Sistem

3.4.1. Perangkat Lunak

Dalam membuat sistem ini, perangkat lunak yang digunakan adalah: (1) OS (Windows 9x/NT4.0/2000/XP) (2) PHP - 4.0.5 (3) MySQL - 3.23.32 (4) *Editplus Text Editor v2.10c* (5) Internet Explorer Version 6.0.2800.1106.xpsp1.020828-1920.

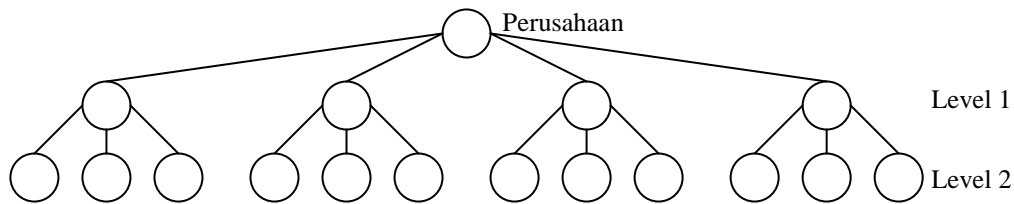
3.4.2. Perangkat Keras

Perangkat keras yang digunakan adalah: (1) Processor Pentium4 1,8GHz (2) DDR 256MB (3) Harddisk 40 GB.

4. Analisa Program

4.1. Analisa Pohon Triner

Pohon triner ini merupakan pohon yang mempunyai subpohon sebanyak 3 buah yaitu subpohon kiri, subpohon tengah, dan subpohon kanan. Pusat (akar) dari pohon dalam MLM ini adalah perusahaan. Jadi, perusahaan ini bisa memiliki subpohon lebih dari 3. Level yang berada di bawah pusat inilah baru menggunakan pohon triner. Untuk lebih jelasnya perhatikan gambar 10.



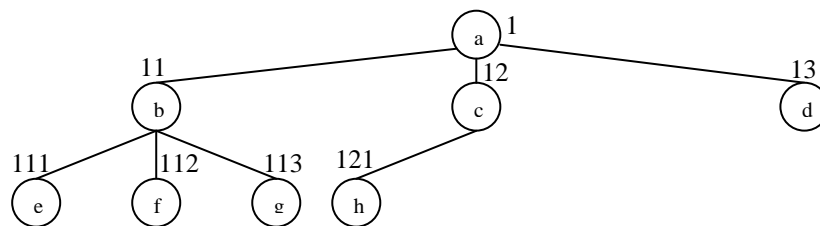
Gambar 10. Struktur Pohon Dengan Akarnya Perusahaan

Dalam penelusuran pohon ini diterapkan algoritma penelusuran level *order*. Untuk penelusuran ini, maka setiap pohon harus diberi identitas dalam penerapan penelusuran level *order*. Identitas ini digunakan untuk menentukan kedudukan anggota dalam struktur pohon. Untuk memudahkan penelusuran, maka identitas ini menggunakan aturan sebagai berikut : (1) Subpohon kiri diberi angka 1 (2) Subpohon tengah diberi angka 2 (3) Subpohon kanan diberi angka 3.

Aturan ini hanya berlaku untuk level 2, yaitu 2 level di bawah perusahaan. Untuk anggota yang berada langsung dibawah perusahaan menggunakan sistem penomoran dengan kenaikan 1. Penomoran ini dimulai dari 1 sampai 9. Jika telah mencapai angka 9, maka nomor akan berubah menjadi huruf A sampai Z. Jika telah mencapai huruf Z, maka nomor akan berubah menjadi huruf a sampai z. Dalam penomoran dengan sistem ini, perusahaan dibatasi hanya memiliki 62 level 1.

Untuk diterapkan dalam suatu basis data, maka diperlukan suatu *field* untuk menampung nomor pohon ini. *Field* ini diberi nama No_Pohon dalam tabel Anggota. Jadi dengan menggunakan *field* ini maka tidak perlu dibuat suatu pointer. *Field* No_Pohon ini sendiri sudah berfungsi sebagai pointer.

Fungsi dari algoritma penelusuran level *order* disini adalah untuk mendapatkan nomor pohon pada waktu seseorang mendaftar yang kemudian akan disimpan dalam *field* No_Pohon pada tabel Anggota. Dalam penerapan level *order* ini membutuhkan suatu prosedur antrian. Namun karena diterapkan dalam basis data, maka prosedur tersebut diganti menjadi pembuatan tabel temporer. Untuk lebih jelasnya perhatikan data pada gambar 11.



Gambar 11. Contoh Data Pohon

Jika ada anggota mendaftar dengan sponsornya adalah 'a' dengan nomor pohonnya adalah 1, maka akan dilakukan *generate* nomor pohon dengan melakukan penelusuran level *order* ini. Sebelum melakukan penelusuran, maka dibuat dulu tabel temporer bernama Antrian sebagai tempat untuk menampung antrian. Setelah itu baru akan dilakukan penelusuran dimulai dari 'a'. Proses dari penelusuran ini adalah sebagai berikut : (I) (1); Isi tabel antrian setelah mengunjungi akar. (II) (1, 11, 12, 13); Isi tabel antrian setelah mengunjungi a. (III) (1, 11, 12, 13); Isi tabel antrian setelah mengunjungi b. (IV) (1, 11, 12, 13); Isi tabel antrian setelah mengunjungi c.

Setiap pengunjungan, akan dilakukan pemeriksaan apakah tempat yang dikunjungi tersebut subpohonnya telah mencapai 3 atau belum. Jika sudah, maka akan dilakukan penelusuran ke tempat lain. Jika belum, maka dari sinilah akan didapatkan nomor pohon yang baru. Dengan data seperti pada gambar 22, pada proses (IV), penelusuran langsung dihentikan karena telah didapatkan nomor pohon yang baru, yaitu 122. Tabel antrian akan langsung dihapus saat nomor pohon telah didapatkan karena tabel ini tidak dibutuhkan lagi.

4.2. Analisa Sistem *Push up*

Perhitungan bonus ini menggunakan sistem *push up*. Dalam perhitungan ini, pertama kali yang akan dilakukan adalah mengecek siapa sajakah anggota yang berhak untuk mendapatkan bonus. Syarat pertama untuk mendapatkan bonus adalah anggota tersebut harus melakukan transaksi pembelian *voucher* minimal 1 kali. Untuk melakukan pengecekan ini,

menggunakan *query* seperti pada gambar 12.

Syarat kedua adalah bawahan dari anggota tersebut juga membeli *voucher*. Setiap jumlah *voucher* itu akan mempunyai perhitungan tersendiri, dimana jika berada pada level ganjil akan dikali Rp. 100, sedangkan pada level genap akan dikali Rp. 500. Namun dalam perhitungan bonus ini, yang akan disimpan dalam tabel hanyalah jumlah *voucher* yang dibeli. Sedangkan hasil dari perhitungan ini tidak disimpan. Hasil perhitungan ini dilakukan dengan program dan langsung ditampilkan.

```
"SELECT No_Pohon, Pembelian.No_Anggota, SUM(Jml_Voucher) " .
"FROM Pembelian, Anggota " .
"WHERE Anggota.No_Anggota = Pembelian.No_Anggota " .
"AND Month(Tgl_Beli) = $Bulan AND YEAR(Tgl_Beli) = $Tahun " .
"GROUP BY No_Pohon, Pembelian.No_Anggota", $koneksi);
```

Gambar 12. Query Untuk Memeriksa Orang Yang Berhak Mendapat Bonus

Sistem *push up* ini akan terjadi jika ada bawahan dari suatu anggota tidak membeli *voucher*, sedangkan anak dari bawahannya itu membeli *voucher*. Hal ini akan menyebabkan anak dari bawahan tersebut menggantikan posisinya untuk perhitungan bonus jaringan ini. *Script* untuk sistem *push up* seperti pada gambar 13.

```
$pohon = $h_q_bawahan[0];
//Cek Apakah Bawahan Membeli Voucher
$cocok = mysql_query(
    "SELECT No_Pohon, Jml_Bonus FROM Pohon_Lama " .
    "WHERE No_Pohon = '$pohon'", $koneksi);
$h_cocok = mysql_fetch_row($cocok);

//Jika Bawahan Membeli
if ( mysql_num_rows($cocok) > 0)
{
    $L[$j] = $L[$j] + $h_cocok[1];
    mysql_query(
        "DELETE FROM Pohon_Lama WHERE No_Pohon = '$pohon'", $koneksi);
}
//Jika Bawahan Tidak Membeli
else
{
    $b_cocok = mysql_db_query("MLM",
        "SELECT No_Pohon, Jml_Bonus FROM Pohon_Lama " .
        "WHERE LEFT(No_Pohon, LENGTH($pohon)) = '$pohon' " .
        "AND LENGTH(No_Pohon) > LENGTH('$pohon') " .
        "ORDER BY No_Pohon");
    if ( mysql_num_rows($b_cocok) > 0 )
    {
        $L[$j] = $L[$j] + mysql_result($b_cocok, 0, "Jml_Bonus");
        $hapus = mysql_result($b_cocok, 0, "No_Pohon");
        mysql_query(
            "DELETE FROM Pohon_Lama WHERE No_Pohon = '$hapus'", $koneksi);
    }
}
}
```

Gambar 13. Script Sistem Push up

Dalam sistem *push up* ini sebelumnya akan dicek terlebih dahulu apakah bawahan membeli atau tidak. Jika membeli, maka jumlah *voucher* yang dibeli akan langsung ditampung dalam variabel $\$L[\$j]$ yang berbentuk array. Jika tidak membeli, maka akan dilakukan pengecekan kembali, apakah anak dari bawahan tersebut membeli *voucher* atau tidak. Jika membeli, maka jumlah *voucher* yang dibeli akan ditampung dalam variabel $\$L[\$j]$. Dalam proses penampungan inilah terjadi sistem *push up* karena $\$j$ akan menunjukkan penggantian posisi. Kelebihan dari sistem ini adalah bahwa setiap anggota tidak akan kehilangan hak bonus jaringannya selama levelnya belum melebihi kedalaman 10 karena setiap bawahan bisa naik menggantikan posisi yang di atasnya. Kekurangannya adalah harus diadakan pengecekan khusus terhadap yang membeli *voucher* sehingga lebih lama.

5. Kesimpulan

Kesimpulan yang dapat ditarik adalah sebagai berikut: (1) Program yang dibangun telah mampu menampilkan visualisasi struktur yang berbentuk pohon dengan menggunakan *script* yang bekerja di sisi *server* dan dibantu dengan *script* yang bekerja di sisi *client* untuk mempermudah tampilan visualisasinya. (2) Program yang dibangun mampu membatasi kedalaman bawahan sampai level berapapun dengan memasukkan kondisi tertentu di klausa *WHERE* pada *query*. (3) Sistem *push up* membuat anggota tidak kehilangan hak bonusnya sampai dengan kedalaman 10 level. (4) Keterbatasan karakter membuat sistem penomoran pohon dalam program yang dibuat ini hanya bisa membuat bawahan yang langsung dari pusat pun menjadi terbatas.

Referensi

- Hariyanto, Bambang. 2003. *Struktur Data Memuat Dasar Pengembangan Orientasi Objek Edisi Kedua*. Bandung: Penerbit Informatika Bandung.
- Shaffer, Clifford A. 1997. *A Practical Introduction to Data Structures and Algorithm Analysis*. USA: Prentice Hall International, Inc.
- Susanto, Budi. 2004. *Databases dan Web*, (Online), (<http://lecturer.ukdw.ac.id/budsus/webdb/week1.pdf>, diakses 28 Agustus 2004)
- Ullman, Jeffrey D., dan Jennifer Widom. *A First Course in Database Systems* Edisi Kedua. New Jersey: Prentice Hall Upper Saddle River.
- Wood, Derick. 1993. *Data Structures, Algorithms, and Performance*. USA: Addison-Wesley Publishing Company, Inc.